2015

Investigation of capabilities and technologies supporting rapid UAV launch system development

Livesay, Patrick Alan

Monterey, California: Naval Postgraduate School

http://hdl.handle.net/10945/48132

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

INVESTIGATION OF CAPABILITIES AND TECHNOLOGIES SUPPORTING RAPID UAV LAUNCH SYSTEM DEVELOPMENT

by

Patrick Alan Livesay

June 2015

Thesis Advisor:	Timothy H. Chung
Second Reader:	Kevin D. Jones

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704–0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE 06-19-2015	3. REPORT TYPE AND DATES COVERED Master's Thesis 09-01-2014 to 06-19-2015	
4. TITLE AND SUBTITLE INVESTIGATION OF CAPABILITIES AND TECHNOLOGIES SUPPORTING RAPID UAV LAUNCH SYSTEM DEVELOPMENT			5. FUNDING NUMBERS
6. AUTHOR(S) Patrick Alan Livesay			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943			8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			10. SPONSORING / MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES The views expressed in this document are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol Number: N/A.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

Unmanned Aerial Vehicles (UAVs) are playing progressively more complex roles in private, commercial, and military applications. One developing mission set of interest to entities in the governmental and defense sectors is UAV swarming: a concept of unit deployment where individuals exhibit complex behaviors when acting as members of a group that are not observed when those units act in isolation. While several barriers exist, one capability gap that must be bridged for fixed-wing systems is the ability to facilitate operationally relevant sortie generation rates. While creating systems mechanically capable of high launch rates is key, there are supporting capabilities that should be considered during the design of UAV launch systems to increase usability and margin to safety. Integration with existing control systems, detection and response to environmental changes, safety interlocks, and software can help achieve these goals and produce a more robust launcher. This report focuses on the identification, selection, and development of such capabilities, which are implemented into launch systems through an iterative prototyping process. Ultimately, a new UAV launch system is created and demonstrated through operational experimentation: one capable of high launch rates, integration with existing control systems, and additional sensors-based capabilities that have heretofore never been seen.

14. SUBJECT TERMS			15. NUMBER OF PAGES
swarm, unmanned aerial vehicle, UAV, launcher, launch system, systems engineering, prototyping, analytical heirarchy process, ROS, Robot Operating System			169
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500 | Standard Form 298 (Rev. 2–89) Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

INVESTIGATION OF CAPABILITIES AND TECHNOLOGIES SUPPORTING RAPID UAV LAUNCH SYSTEM DEVELOPMENT

Patrick Alan Livesay
Lieutenant, United States Navy
B.S., United States Naval Academy, 2008

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 2015

Author: Patrick Alan Livesay

Approved by: Timothy H. Chung
Thesis Advisor

Kevin D. Jones
Second Reader

Clifford A. Whitcomb
Chair, Department of Systems Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Unmanned Aerial Vehicles (UAVs) are playing progressively more complex roles in private, commercial, and military applications. One developing mission set of interest to entities in the governmental and defense sectors is UAV swarming: a concept of unit deployment where individuals exhibit complex behaviors when acting as members of a group that are not observed when those units act in isolation. While several barriers exist, one capability gap that must be bridged for fixed-wing systems is the ability to facilitate operationally relevant sortie generation rates. While creating systems mechanically capable of high launch rates is key, there are supporting capabilities that should be considered during the design of UAV launch systems to increase usability and margin to safety. Integration with existing control systems, detection and response to environmental changes, safety interlocks, and software can help achieve these goals and produce a more robust launcher. This report focuses on the identification, selection, and development of such capabilities, which are implemented into launch systems through an iterative prototyping process. Ultimately, a new UAV launch system is created and demonstrated through operational experimentation: one capable of high launch rates, integration with existing control systems, and additional sensors-based capabilities that have heretofore never been seen.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

AE All Environment

AHP Analytic Hierarchy Process

AMPPS Automated Multi-Plane Propulsion System

API application program interface

ARSENL Advanced Robotic Systems Engineering Laboratory

AWG American Wire Gauge

CAD computer aided design

CBA Capability Based Assessment

CONOPS Concept of Operations

DAG Defense Acquisition Guidebook

DOD Department of Defense

EW electronic warfare

GCS ground control station

GPS global positioning system

ICD Initial Capabilities Document

ISR intelligence, surveillance, and reconnaissance

JATO jet assisted take-off

JCA Joint Capability Area

JCIDS Joint Capabilities Integration and Development System

LCD liquid-crystal display

LED light-emitting diode

NPS Naval Postgraduate School

RATO rocket assisted take-Off

RFID radio frequency identification

ROS Robot Operating System

RULE Rapid UAV Launch Engine

SAR search and rescue

SEAD suppression of enemy air defenses

U.S. United States

UAS unmanned aerial system

UAV unmanned aerial vehicle

USB universal serial bus

VTOL vertical take-off and landing

Executive Summary

This research was an examination of the processes needed to design, build and test rapid-cycle UAV launchers to support the deployment of swarming UAV systems. The primary goal for the work completed in support of this project was to develop a launch system for fixed-wing UAVs that was easily transportable, straightforward to operate, and was capable of very short launch-cycle times. More specific to this particular research effort was the identification, prioritization, selection, and implementation of enabling electrical, software, and sensors-based capabilities that led to increases in the launch system's efficiency, usability, and margin to operator safety. Ultimately, the launch system design team was able to develop a set of prototypes that exhibited varying, yet generally expanding degrees of capability, culminating in the creation of a revolutionary launch-system for fixed-wing UAVs.

The prototyping process began through the development of a clear understanding of the context and environment in which the new launch system was expected to operate. This enabled the launcher design team to more clearly determine and articulate system requirements and performance parameters. Next, a spanning set of likely operational scenarios were defined and, from these scenarios, a comprehensive list of potential launch-system capabilities were identified. These capabilities were then mapped to their corresponding Joint Capabilities Integration and Development System (JCIDS) Joint Capability Areas (JCAs) for future ease of reference for military-specific applications of these capabilities and technologies.

Capability priority metrics were established to facilitate the prioritization and organization of these potential capabilities. For this effort, the three metrics selected were the number of operational scenarios to which the capabilities would likely contribute, an overall estimate of the utility provided by the capability to the launch process, and an estimated degree of difficulty associated with the implementation of each capability. Nominal, minimum, and maximum values were then assigned to each individual capability for each metric by the design team in what is, admittedly, a somewhat subjective process. To account for the subjective nature of these score assignments and, in part, due to the large number of potential capabilities identified, an Analytic Hierarchy Process (AHP) was performed to

prioritize the capabilities and assist in the decision-making process [1].

The AHP decision-analysis technique is best suited to situations involving large numbers of alternatives when multiple criteria are being used to evaluate the alternative options [1]. In this method, capabilities are compared head to head against all others taking into account only one decision-criteria at a time [1]. The results of these comparisons are then weighted using weighting values determined through a similar procedure, and these weighted capability scores are used to prioritize the alternatives [1]. For this effort, the entire AHP process was repeated multiple times to ensure a robust set of data was collected before averaging the resultant scores for each capability to produce the final score set. All the capabilities were then ranked based on these final, average scores. Finally, natural gaps in the capability scores were identified, and groups of potential capabilities were designated for implementation into the various launch-system prototypes.

The first launch system prototype was the Rapid UAV Launch Engine (RULE), which utilized a tank of compressed air, a solenoid-operated, three-way pneumatic valve, and a pneumatic actuating cylinder as the means of propelling a UAV mounted at the opposite end of a lever arm and pivot assembly [2]. The system, shown in Figure 1 also leveraged a laptop computer running Linux and the Robot Operating System (ROS) to control software-side functions and facilitate more efficient system operation [2]. Enabling capabilities originally identified for implementation into this prototype were:

1. Abort launch functionality
2. Mechanical-based kill switches with easy accessibility
3. Moved and setup by one to two technicians
4. Lighting system to warn personnel of launch status
5. Automatic reset capability
6. Launch platform position sensors

Unfortunately, the RULE fell short in its primary directive: launching UAVs at speeds sufficient to sustain temporary flight [2]. The RULE also suffered from other issues, such as poor reliability during full speed operation, poor mobility during operation, and poor system range due to AC power requirements [2]. However, the system did succeed in demonstrating, at a low level, the value that an automatic reset capability could provide to

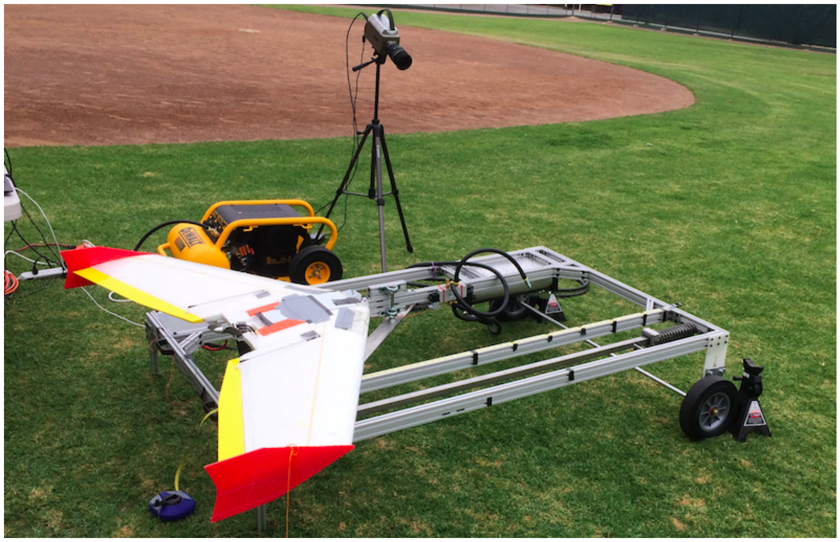

Figure 1: RULE launch system prototype

an operator engaging in rapid UAV launching operations [2].

The second prototype system was the Chain Launcher, which used a bank of four lead-acid batteries and, eventually, a DC motor controller to provide power to a large DC motor connected to a roller chain and sprocket assembly. Once again, the prototype's functionality was controlled through a laptop computer, a wireless game controller, and several USB connections to the key system components. For reference, the prototype is shown in Figure 2. Enabling capabilities identified for implementation into this second prototype were:

1. First-level capabilities not fully implemented or requiring significant design changes from first prototype
2. Safe to load indication
3. Detect environmental parameters (wind data)
4. Maximize launcher range envelope from Ground Control Station
5. Detect people/objects in launcher vicinity
6. Detect UAV on launch platform
7. Disable launch capability if winds averse

In most respects, the Chain Launcher design was considered to be a success. It was able to

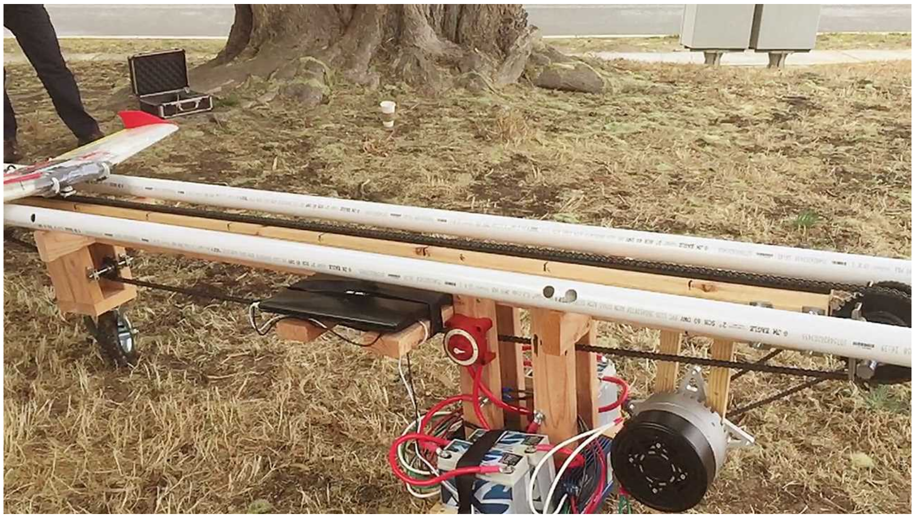
Figure 2: Chain Launcher system prototype

be easily set up, was capable of accelerating and releasing the primary stakeholder's UAV at the desired launch speed, and was able to be configured for an automated reset through the use of software and precisely timed motor commands. The Chain Launcher's design also necessitated an emergent requirement for a powered-wheel subsystem with a wireless external controlling device. However, even with these new capabilities successfully integrated into the design, the system was still not all that a UAV launcher should be. It was hastily built, hastily wired, and lacked many of the enabling capabilities that should theoretically have been implemented and included in this prototype iteration.

Finally, development work commenced on the final prototype, the Automated Multi-Plane Propulsion System (AMPPS). This prototype, shown in Figure 3 was functionally similar to the Chain Launcher that came before, but included a number of refinements and additional capabilities that would have been nearly impossible to incorporate into the Chain Launcher's original design configuration. The AMPPS used motor controllers to precisely operate both the the chain-drive and wheel motors, enabling a highly controlled acceleration profile during launch and a computer-timed reset capability, and also provided for wireless, stand-off control-ability. Enabling capabilities identified for implementation into the final AMPPS prototype were:

1. First and second-level capabilities not fully implemented in first or second prototypes

2. Disable launch ability if area unsafe
3. Streamline setup and initialization
4. Communicate launch system/sensor status to the ground control station (GCS)
5. Receive “Halt Launch” commands from the GCS or safety observers
6. Re-orient launcher if wind direction not favorable
7. Disable launch ability until UAV loaded

Figure 3: Automated Multi-Plane Propulsion System prototype

Ultimately, the overall design and implementation of the AMPPS launch system was considered to be a resounding success. It was even easier to setup than the Chain Launcher, provided for standoff mobility using the powered wheels and wireless gamepad interface, controlled the acceleration profile of launched aircraft with unheard-of accuracy and consistency, and was capable of automated reset through software-based timing functions. The AMPPS also alerted the operator of personnel in the launch path, wind conditions inconsistent with the launcher’s orientation, and could automatically identify the specific aircraft loaded onto the launcher interface and communicate this information to the launch technician or onboard camera systems. In field testing, the system executed more than twenty successful launches, with only one anomalous launch attributed to a flaw in the specific aircraft’s construction. As shown in Figure 4, testing results for the AMPPS indicated a generally well-designed and well-constructed rapid-launch system with significant poten-

tial for getting large numbers of lightweight, fixed-wing UAVs in the air.

Figure 4: AMPPS prototype demonstrating a successful UAV launch

List of References

[1] C. Ragsdale, *Spreadsheet Modeling and Decision Analysis*, 5th ed. Mason, OH: Thompson South-Western, 2008.

[2] G. Viera, J. Webb, D. Cummings, P. Livesay, R. Davis, J. Ng, and A. Abulfateh, "Rapid unmanned aerial vehicle launch engine design capstone report," M.S. capstone report, Dept. Systems Eng., Naval Postgraduate Sch., Monterey, CA, 2014.

Acknowledgments

I would first like to thank my advisor, Dr. Tim Chung, for his guidance, support, and confidence in my abilities and the ultimate success of this project. His general hands-off approach, while still assisting and, when needed, intervening at key points throughout the process, were instrumental to facilitating the environment needed for my partner and I to succeed in this ambitious effort. The successful completion of the AMPPS launcher is both a reflection and a result of his vision, expertise, and willingness to let us succeed (or fail) on our own terms. He is also one of the best leaders I have ever had the opportunity to work for and his dedication to his students, his ARSENL team, and to their mission has provided an outstanding example that I hope to emulate throughout my own career.

Next, I would be remiss if I didn't thank my partner in the launcher-development effort, LT Raymond Davis. As the mechanical lead, he spent countless hours and late nights developing CAD drawings for the prototypes, brainstorming UAV attachment methods, and facilitating electronics and sensors integration. The ultimate success of this project is due largely to his expertise and dedication, and I am eternally grateful that he convinced me that this was a project worth taking on.

I'd also like to specifically thank my second reader, Dr. Kevin Jones, for his support during this process. Dr. Jones provided critical feedback and data regarding the capabilities and requirements of the Zephyr aircraft during nearly every stage of the design and construction of the various prototypes. The time he spent in meetings with us or at the softball field conducting testing, not to mention time spent reviewing and editing my thesis, are very much appreciated.

Several other members of the ARSENL team also deserve acknowledgment for their contributions to this effort. Mike Clement provided critical support regarding software development and implementation throughout the process, and I can honestly say that the AMPPS would not have been completed without his expertise and assistance. Marianna Jones, who helped us navigate the treacherous waters of the government purchasing system by placing and monitoring the status of all parts orders for us, also deserves many thanks for all the time and effort she invested. In doing so, she saved us both many hours, enabling us to

focus more on accomplishing the primary mission and less on the administrative burden. Finally, input and feedback from the other ARSENL team members, specifically Michael Day and Duane Davis, was key to facilitating good software integration with existing systems.

Finally, I'd like to thank my wife, Haley, my son, Jackson, and both of our families for the unwavering patience, understanding, and support they all continue to provide. I'm fortunate to have a wife who understands my inability do anything half-way and is willing to accept the hardships that often arise as a result. She has essentially been a single parent to our son as I worked on construction, testing, and writing these past months, and her willingness to do so without complaint is an amazing example of the same support she's provided since we were only teenagers. I am truly unworthy of the selfless, unconditional love and support that she has always provided. Additionally, I'd like to thank our families, and especially my parents, for the parts they played in fostering my many, widely varying interests and allowing me the freedom to explore them from a young age. The support and understanding they've shown over the years, and especially over the past seven years of my naval career, is frequently recognized and much appreciated. It is this ever-present support base that has enabled Haley and me to be successful in so many of our personal and professional endeavors. Last, I'd like to dedicate my portion of this effort to the memory of my grandfather, my family's first Engineer, who helped instill in me both a thirst for knowledge and an interest in how things work, and who showed a deep interest in the development of this launch system from its inception.

CHAPTER 1:
Introduction

1.1 Motivation

Over the past few decades, unmanned aerial vehicles (UAVs) have played ever-increasing and progressively more complex roles in private, commercial, and military applications. These aircraft, which can be designed as traditional rotary wing, multi-rotor, or fixed-wing platforms, are powered aerial vehicles that do not carry a human operator and are capable of flight with or without human remote control [1]. They can be expendable or recoverable, and are capable of performing highly diverse and increasingly complicated mission sets, such as intelligence, surveillance, and reconnaissance (ISR), electronic warfare (EW), suppression of enemy air defenses (SEAD), and remote strike activities [2]. In non-military applications, UAVs are frequently deployed in search and rescue (SAR) missions, monitoring of electronics and communications grids, agricultural crop monitoring, and meteorological assessments or traffic surveillance activities [3].

1.1.1 Military Necessity of Unmanned Aerial Vehicles

Due in large part to their highly successful employment in the United States' recent warfighting efforts in Iraq and Afghanistan, UAVs have, in recent years, begun assuming roles and performing missions that were formerly assigned only to manned aircraft [4]. The use of these remotely operated UAVs in place of manned aircraft provides the organization with three distinct advantages. First, they eliminate any risk to the pilot's life that would have been assumed during the execution of a manned mission [4]. The ability to provide first-rate intelligence gathering, command and control, targeting, and weapons delivery while reducing risks to the war fighters and decreasing the likelihood of casualties is, by far, the most desirable feature of unmanned aerial systems (UASs) [5]. Second, the development and procurement costs of most UASs are significantly lower than those associated with manned aircraft and support systems [4]. This concept of minimizing cost while maximizing the capabilities of these platforms remains at the forefront of the UAV conversation today, especially as defense budgets continue to shrink in the current post-war political climate. Finally, due in large part to recent technological advancements in communications

and sensor array capabilities, many UAV platforms are capable of traveling significantly farther and remaining on station longer than manned aircraft or satellite surveillance systems currently allow [3].

The utilization of these unmanned systems also comes with several disadvantages. First, the accident rates for UAVs are significantly higher than those associated with manned-aircraft operations [6]. This is, in part, because most current UAV systems are intentionally designed with fewer system redundancies and backup systems in order to minimize costs [7]. The justification for this less robust system design is that, since the aircraft are unmanned, the total cost associated with losing a unit is significantly reduced as compared with a scenario where the life of a highly trained human operator is on the line. Additionally, since the pilots operating these aircraft are physically removed from the system, they are less equipped to properly identify and take corrective action on problems that arise during flight operations [7]. Another major disadvantage is that, while UAVs are cheaper to develop and initially procure, they are also significantly more expensive to operate due to their extensive logistical support, specialized maintenance, and operator training requirements [8]. In order to make future UASs more cost effective, many aircraft and systems currently in development are being designed to operate much more autonomously with control stations that allow a single operator to control multiple UAVs simultaneously [3], [7].

Despite these issues, today's rapid pace of technological development has fostered a high level of confidence in the future of UAV programs and capabilities across many governmental, commercial, and private organizations. In fact, the Teal Group, an aerospace and defense market analysis firm based out of Fairfax, Virginia, recently declared UAVs to be the "most dynamic growth sector of the world aerospace industry this decade" and projected that worldwide spending on UAVs and their support systems will reach or exceed $89 billion over the ten-year period beginning in 2012 [9]. The United States (U.S.) government seems to agree. In fact, the Department of Defense's (DOD's) fiscal investment in UAV research and procurement has risen from $667 million in FY2001 to nearly $3.9 billion in FY2012 and, during that time, their arsenal of aircraft grew from 167 to approximately 7,500 [4], [10]. This represents a 500% increase in spending and a 4,400% increase in aircraft inventory over a period of only 11 years. Further, the U.S. is not alone in its

interest in the unique capabilities and benefits that can be provided by these unmanned systems. Commercial and governmental organizations in Europe (France, Germany, Italy), the Middle East (Israel, Iran), Asia (China, India, Japan, South Korea), and Russia are all actively working to develop new unmanned aerial systems and strategies for deploying them [3]. Recognizing the rapid pace of worldwide technological development, along with the volatile nature of the current geo-political climate and the increased focus on fiscal responsibility, the DOD's goal to be the "most innovative user" of these systems becomes all the more important [9].

1.1.2 Swarm Mission

One developing mission area that is of particular interest to many entities in the defense and commercial sectors is that of UAV swarming. Swarming is a concept of unit deployment that is based largely upon the observations of emergent behaviors in the natural world. Specifically, wolves, flocking birds, and insects (e.g., ants and bees) have demonstrated the ability to conduct complex behaviors when acting as members of groups that are not observed when individual members act in isolation [11]. Most importantly, although the groups almost always appear to be highly organized, there is a noted absence of supervisory behavior in these systems [12]. Instead, it seems that for the majority of the actions and tasks completed by these groups, the coordination of labor that emerges stems largely from the interactions and cooperative efforts among individual units [12]. It is this highly organized yet largely decentralized set of emergent behaviors that many of today's researchers and military policy-makers are hoping eventually to replicate using unmanned robotic systems.

These nature-inspired swarming tactics have been adapted for use in human warfare applications on multiple occasions throughout history. Arquilla [13] defines swarming as a "seemingly amorphous, but... deliberately structured, coordinated, strategic way to strike from all directions, by means of a sustainable pulsing of force and/or fire, close-in as well as from stand-off positions." He goes on to provide multiple examples of the use of swarm tactics throughout the history of human conflict, such as the Greeks in their naval victory over the Persians at the Battle of Salamis, the Mongols during their attempted invasions of Japan in the 13th century A.D., and the British Navy in their battle against the Spanish Armada in the late 1500s. Both Edwards [11] and Shannon [14] also identified other examples of the

use of swarm tactics throughout history, including American colonists fighting the British during their march from Lexington, Massachusetts during the American Revolution, Chinese light infantry fighting against the First Marine Division at the Chosin Reservoir during the Korean War, and Islamic Jihadists fighting against American ground forces in Baghdad during Operation Iraqi Freedom. It is worth noting that the tactical element common to all of these engagements is that the swarming forces repeatedly converged on their targets from multiple directions, executed rapid, pulsed attacks, and then re-dispersed in an effort to minimize losses while confusing and destroying the enemy [11].

Applying the fundamental principles of swarm theory to unmanned robotic systems could provide the user with a number of intriguing benefits. The hope is that, through the development of an artificial "swarm intelligence" [12], ground and aerial-based robotic systems could eventually be used to execute battles against enemy forces with an unprecedented degree of autonomy. As previously discussed, one key attribute affecting the costs of most unmanned aerial vehicles is the number of personnel required to operate the vehicle and its associated support systems. The development of the aforementioned swarm intelligence would be a key step in creating a system of multiple UAVs that can be operated simultaneously by a single individual, making unmanned aerial systems much more cost-effective. Additionally, a swarm of UAVs, acting as a highly integrated network of dispersed assets, could likely enhance the separate (and potentially unique) capabilities of individual units to more effectively execute dangerous, dull, or politically sensitive missions that have traditionally been reserved for manned aircraft or larger, more expensive UAVs [3]. These concepts pave the way for an even more intriguing idea. A fleet of inexpensive, swarm-capable UAVs could provide a unique redundancy advantage in addition to saturating enemy threat-detection sensors and weapon systems [3]. If this fleet were established with the same collective capabilities as one of the highly capable UAVs currently in the DOD's arsenal, these advantages could likely facilitate mission completion with significantly reduced financial risk due to lost or damaged units.

Arquilla [13], in "Swarming and the Future of Conflict," postulates that a paradigm shift toward the use of swarm tactics and other means of non-linear warfare using unmanned systems is on the not-so-distant horizon [13]. If the U.S. hopes to facilitate this shift through the use of swarm-capable UAVs, there remain a number of obstacles that must first be over-

come. Miller [3] identifies collision avoidance between individual elements and the ability to keep the swarm on its assigned mission until completion as two of the most significant challenges to be overcome. Clough [15] poses additional, more theoretical questions such as how to change swarm behavior in real-time, how to ensure the swarm units only attack enemy targets and not friendly forces, and how to organize behavioral tendencies in individual units in a way that enables them to efficiently switch between individually dictated and swarm guided actions based on sensory inputs. Interestingly, however, both these authors and others fail to identify one important obstacle that has yet to be solved regarding swarm UAV capabilities: how can an organization safely and efficiently get a large number of these aircraft airborne in a short period of time?

1.2 Problem Identification

1.2.1 Aircraft Platforms and Limitations

One potential solution for launching many UAVs quickly is to use rotary wing aircraft or platforms with vertical take-off and landing (VTOL) capabilities. Helicopters, quadcopters, and other rotary wing platforms are ideal for applications where high degrees of maneuverability or the ability to hover and loiter for periods in a specific area are necessary. However, while they do excel in certain applications, rotary wing aircraft have certain limitations that could make them less appealing for employment in a swarm attack scenario. For instance, the configuration of the rotors on these aircraft tends to limit their overall aerodynamic efficiency during flight, resulting in lower top speeds and more limited ranges than comparable fixed-wing platforms. Additionally, rotary wing aircraft tend to have a higher degree of mechanical complexity inherent in their designs, which can make them more expensive to produce and maintain than comparably equipped fixed-wing platforms. Conversely, fixed-wing aircraft are generally capable of higher top speeds, longer mission ranges, and large payload ratios, favorable characteristics that can facilitate attacks on more distant targets, earlier interception of incoming attacks, and can make the aircraft more survivable than their rotary-wing counterparts [16].

Hybrid solutions, such as aircraft like the MV-22 Osprey or the new Joint Strike Fighter, have recently been designed to have VTOL capabilities that enable a vertical, rotary-wing style takeoff before transitioning to fixed-wing style flight and maneuverability. Similarly,

other aircraft have been designed with rocket assisted take-Off (RATO) or jet assisted take-off (JATO) capabilities where rocket or jet engines are used to initially accelerate the aircraft for takeoff, enabling a significantly shorter runway or launch space to be used while still facilitating the advantages provided by fixed-wing platforms during normal flight [17]. While these innovative systems are designed to combine the benefits provided by fixed and rotary-wing flight platforms into one highly capable aircraft, they are not without their own disadvantages. First, these hybrid systems tend to be significantly more complex than traditional rotary or fixed-wing platforms in terms of both the mechanical design and the operation of the aircraft. These complexities drive up the costs of acquisition and operation, and will require operators who are specifically trained to operate the unique feature sets of the aircraft. Additionally, the use of explosive propellants when employing RATO or JATO systems both drives up the cost of launching each aircraft and creates a situation where the safe storage and transport of these fuels becomes an issue, adding a myriad of cost, safety, and regulatory compliance concerns. Furthermore, RATO or JATO systems require a buffer area of approximately 100,000 square feet in the launch path to ensure adequate margin to safety when the explosive propellants are ignited during launch [17]. Finally, all these alternative launch methods put some parts of the aircraft under forces which are highly unusual for standard fixed-wing platforms and may necessitate additional analyses of the structural reliability of the aircraft during launch.

As discussed earlier, one of the likely benefits of using a swarm of UAVs to carry out an aerial attack (or defensive) scenario is the redundancy advantage that could be provided at relatively low cost. These redundancies could create a situation where the attack could easily continue despite the losses of a few individual units. Since each unit is more specialized and less expensive to build than most modern, highly capable UAV systems currently in the DOD's arsenal, the loss of a single unit, although significantly more likely, would represent a mere fraction of the cost of losing a highly equipped aircraft such as the Predator, Shadow, or Global Hawk. Furthermore, due to capability redundancies designed into the various aircraft engaged in the swarm, the loss of even a few units would likely represent a small, potentially insignificant loss in overall operational capability. Given that a limited number of these unit losses would generally be expected during an offensive or defensive scenario where swarm UAVs are utilized, the cost savings provided by creating this swarm from a significantly cheaper, minimally complex fixed-wing platform could be

quite significant over time. Additionally, since fixed-wing platforms are generally capable of higher top-speeds and more distant ranges than rotary wing aircraft, individual aircraft and their associated support systems may benefit from increased levels of survivability due to increased maneuverability and better isolation from the enemy.

At this point, it should be readily apparent that numerous solutions for launching a swarm of UAVs in a given period of time already exist. These solutions can be relatively simple, such as utilizing a fleet of traditional rotary-wing aircraft like helicopters or quad-copters, or more complex, as would be the case when using fixed-wing aircraft with VTOL and RATO capabilities. Ultimately, each of these platforms and launch solutions comes with its own set of advantages and disadvantages. However, for the purposes of this study, the design of potential UAV launch systems is focused toward creating a swarm of traditional, fixed-wing aircraft without using VTOL or RATO systems. The primary drivers for implementing these scope limitations are to minimize initial aircraft acquisition costs, reduce the cost of unit losses which might occur during an engagement, and to minimize the overall mechanical complexity and training required to operate the UAVs. Most importantly, the ability to safely and efficiently launch large numbers of traditional fixed-wing UAVs in a very short period of time presents a capability gap that has yet to be bridged in both the public and commercial sectors.

1.2.2 Re-thinking the UAV Launch System

There are many obstacles that still need to be overcome before an organization is able to fully implement a large-scale battle of swarming UAV forces. For instance, UAVs need to be capable of identifying and tracking the positions and flight paths of both themselves and other units with a much greater degree of precision than current low-cost global positioning system (GPS) technologies can independently facilitate [18]. Additionally, keeping the UAVs from crashing into one another while still assembling and flying in tactical formations and ensuring that they stay on mission until that mission has been verified complete are challenges that have yet to be overcome in the area of swarm research [3]. However, the ability to deliver operationally relevant sortie generation rates using a minimal number of launch technicians is also an important capability gap that is rarely identified as a UAV swarm operational limitation. Nevertheless, the efficient launching of these units is a critical aspect of future swarm UAV operations that must be solved.

While creating a system mechanically capable of facilitating high UAV launch rates is a paramount concern, there are numerous software and sensors-based capabilities that could be implemented into the design of a UAV launch system that would significantly enhance the system's utility in the field. The first such capability would be effective communications and integration with existing UAV operational control systems. This integration ensures that the ground control station (GCS) operator(s), the "Swarm Commander," and any other personnel involved in the supervision or operational control of the UAV fleet have adequate situational awareness with regard to the status of the launch platform, UAVs being prepared for launch, and any launcher support systems that place limitations on the launcher. Additionally, a straightforward launcher interface and control system would help to ensure that only a minimal number of launch technicians are required to execute launches. Such a launcher could also be equipped with safety features that prevent inadvertent or unexpected launch actuation and ensure that all personnel are adequately clear of the launch area and UAV flight path prior to initiating a launch. Finally, many organizations would likely desire that the launch system be designed for maximum reliability to mitigate burdensome maintenance requirements, potential repair costs, and the likelihood of system failure during operation.

1.3 Benefits of the Study

The intent of this work is to identify, prioritize, develop, and test technologies that facilitate the operation of a rapid UAV launching system in support of swarm UAV flight operations. Rapid launching capabilities for small, fixed-wing UAVs are a newly emergent requirement and, as such, require unique software and integration solutions. As such, a comprehensive set of potential feature sets that could be included in a rapid UAV launch system are identified in this work, and a means of prioritizing those capabilities and making design decisions during an iterative prototyping process are discussed at length. Specifically, "smart" technologies are introduced which provide increased functionality, usability, and safety to both the launch system mechanical design and the system operator.

All launch system and capability development efforts are specifically constrained by the limitations posed by the Zephyr II flying-wing aircraft and associated support systems currently being utilized by Naval Postgraduate School's (NPS) Advanced Robotic Systems Engineering Laboratory (ARSENL) research group in their UAV swarm experimentation

efforts. However, the insights gained from this study will be applicable to both commercial and military launch systems as swarm UAV capabilities continue to develop and evolve. Ultimately, this work culminates in the development and demonstration of an innovative UAV launch system with an integrated suite of capabilities that serves not only as a baseline for future launch system development, but also as a milestone in support of fixed-wing swarm UAV research. Potential extensions of this work involve the identification of new capabilities and feature sets not identified herein and the implementation of these new features into an operational rapid UAV launch system.

1.4 Thesis Organization

Having identified the existing capability gap for a rapid-UAV launch system tailored specifically for the deployment of fixed-wing aircraft, the next chapter identifies several fixed-wing launch solutions for UAVs currently employed by the DOD, their limitations, and the scope of launch solutions and potential capabilities proposed through this work. Chapter 3 contains a walkthrough of key portions of the Joint Capabilities Integration and Development System (JCIDS) process as it applies to initial capability selection and development, a set of expected operational scenarios for launch operations, and a means of prioritizing and selecting capabilities for development and implementation. In Chapter 4, an overview of the first UAV launch system prototype developed as part of this effort is provided along with a justification of design decisions made in support of the first round of software, hardware, and electrical-based capability enhancements. Chapter 5 presents an overview of the second launch system prototype and highlights the design and implementation of the second round of software and electrical-based capabilities. Chapter 6 follows a similar pattern, discussing the development and implementation of capabilities and feature sets into the final launch system prototype. Finally, Chapter 7 contains a summary of work completed through this effort, conclusions drawn from development and testing of the prototypes, and recommendations for future work in the UAV launch system domain.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Existing Solutions and Limitations

2.1 Capabilities of Existing Launch Systems

While no organizations have, as of yet, demonstrated an ability to launch large numbers of fixed-wing UAVs (50 or more) in less than ten minutes using only one or two launch technicians, there have been many noteworthy systems created with the goal of accelerating fixed-wing UAVs for flight [9]. Several of these launch systems are identified and discussed herein, with specific emphasis on their limitations in terms of rapid-launch operation for multiple units and their overall lack of integration with outside systems.

2.1.1 Hand Launch

The first UAV launch system that merits discussion is not really a system at all. Instead, an operator launches the aircraft by simply throwing it into the air by hand. This is done by either holding the UAV at a wingtip and spinning in a circle to increase velocity prior to release, or by grasping it on the nose or at the bottom of the fuselage, holding it high overhead, and throwing it into the air at a pre-determined launch angle as shown in Figure 2.1 and Figure 2.2. For these launches, the UAV's propulsion system can either be operating prior to release or can be activated by accelerometers or other sensors onboard the aircraft after the launch has occurred. There are currently several UAV platforms owned and operated by DOD organizations that employ this launch method, including the Wasp All Environment (AE) UAV [19], the RQ-20A Puma AE UAV [20], and the RQ-11 Raven [21].

The primary advantage of the hand-launching method is that no additional equipment, other than the GCS and UAV itself, is required to perform a mission. However, there are several downsides to hand launching these UAVs. First, the aircraft must be specifically designed and configured to facilitate a hand launch. This places limits on the UAV size, weight, configuration, sensor suite, and payload. The operator must also ensure that the UAV is thrown forcefully enough to accelerate the aircraft to its minimum required speed for flight; otherwise, the likelihood of a crash immediately following launch is extremely high. Additionally, since the UAV's propeller might be turning prior to launch, there are safety

Figure 2.1: Hand launch of Army's Puma UAV, from [20]

Figure 2.2: Hand launch of USMC's Wasp AE UAV, from [19]

concerns that should be considered. The operator could accidentally contact a wing, tail, or worse, a spinning propeller, while throwing the UAV into the air. Depending on the speed and torque of the propeller and overall strength of the aircraft's body, this contact could result in both injury to the operator and a crash immediately following launch. Finally, in a situation where the operator desires to execute swarm UAV operations, the use of hand launching would most likely require large numbers of additional personnel on hand to specifically assist in getting the UAVs airborne. This large group of people could be infeasible or cost prohibitive and would likely remove human resources from other, potentially more important jobs during launch.

2.1.2 Conventional Runway Launch

The second UAV launch system that should be identified is, once again, not much of a system in the technical sense. Taking inspiration from the hundreds of fixed-wing aircraft of all shapes, sizes, and configurations that fly all over the world every day, a large proportion of the fixed-wing UAVs currently in operation are designed to takeoff by simply accelerating down a runway until sufficient lift is generated to enable flight. As mentioned previously, this takeoff process can be accelerated and enhanced through the use of external propulsion systems which facilitate a JATO or RATO-style launch. Current UAV platforms in the DOD's arsenal that utilize this launch method include the Air Force's RQ/MQ-1 Predator [22], the Army's MQ-5 Hunter [23] (shown in Figure 2.3), the Navy's MQ-4

Triton [24], and the Air Force's MQ-9 Reaper [25].

Figure 2.3: Conventional runway launch of Army's MQ-5 Hunter UAV, from [23]

The first advantage of utilizing a conventional runway launch its relative simplicity as compared to other launch methods. Since the launch only requires that each UAV be equipped with landing gear of some type, this launch method is both highly reliable and very cost efficient. Additionally, since runway takeoffs are, by far, the most common means of getting manned fixed-wing aircraft airborne, most pilots are very well rehearsed in the procedural requirements, and there are even automated systems that are designed to execute these takeoffs with little human oversight. The primary disadvantage of using a conventional runway takeoff is the requirement for a long, smooth, flat runway. This requirement means that UAVs designed to utilize this launch method are significantly less flexible in terms of acceptable launch locations. For instance, it would be very difficult to launch these UAVs in mountainous regions or areas of rough or rocky terrain. Additionally, since long runs down the runways are needed to generate the lift required for takeoff, it would take a great degree of careful planning and coordination between various aircraft and pilots to facilitate the massive takeoff event required to engage in swarm operations. Finally, the fuel (or battery power) expended while accelerating the aircraft down the runway results in a shorter available flight time for the aircraft. While this difference in available flight time may be fairly small, for many inexpensive UAV platforms, which run solely on battery power, the total available flight time on a full charge is on the order of only 30 to 45 minutes. Given this relatively tight flight window, a five minute reduction in availability is certainly an operationally relevant amount of time which could be restored by utilizing alternative launch

methods.

2.1.3 Tube Launch

The next noteworthy UAV launch system is the tube launcher. Tube-launched UAVs are generally designed with spring-loaded retractable wings and tailpieces that fold in towards the UAV's fuselage. This allows the UAV to fit into a tube which are used to guide the aircraft as it is accelerated to launch speed due to some external phenomenon. This phenomenon usually takes the form of a controlled explosion or munitions detonation. Once the UAV is forced from the tube, the retractable wings and tail assemblies deploy and snap into place and the UAV's own propulsion system is actuated, enabling it to commence controlled flight operations. UAV platforms currently used by DOD organizations which employ this launch method include the Army's AeroVironment Switchblade UAV [26], which is launched using a 70mm rocket launcher and is shown in Figure 2.4, and the Navy's experimental XFC UAV [27], which has been deployed from the torpedo tubes onboard submarines while submerged and is shown in Figure 2.5.

The utilization of a tube launching system for a UAV fleet provides an organization with several unique advantages over other launching techniques. First, the ability to collapse or fold-in the wings and tail assemblies makes the UAVs more transportable. This means that a single person could carry more UAVs to the desired launch location unassisted than would be possible with normal fixed-wing aircraft. The nature of the tube launch system also makes it possible to launch these aircraft from nearly any type of terrain or location since the UAV is accelerated and directed out of the tube and, since the system is re-loadable by simply replacing the mortar or rocket charge and adding a new UAV, many aircraft could be easily launched in a very short period of time. The downside of utilizing a tube-launch system is that the size, configuration, and payload of the UAVs are limited by the size of the launch tube. There are also reliability concerns inherent in the design of tube-launched UAVs since the wings and tail assemblies are mechanically deployed following launch. If these parts fail to actuate properly, the UAV will crash soon after launch. Furthermore, many of the UAVs that currently utilize this launch method are designed to be expendable; they deliver their payload by flying into their targets, resulting in its own destruction as part of its effect. This may possibly increase the overall cost of this system since each UAV launched can only be used to perform a single mission. Finally, there are significant safety

Figure 2.4: Tube launch of Army's Switchblade UAV, from [26]

Figure 2.5: Underwater tube launch of an XFC UAV from a U.S. submarine, from [27]

concerns associated with the transport and use of the munitions or explosives required to accelerate the UAVs out of the tubes. These concerns must be mitigated through the use of administrative requirements and operator training that adds to the cost and overall burden associated with utilizing the tube-based launch method.

2.1.4 Catapult Launch

The final UAV launch system type that merits discussion is the catapult launch system. In these systems, compressed air or high pressure fluid is forced into one end of an actuator, generating either linear or rotary motion that is used to accelerate the UAV. Due to the nature of this design, simple machines such as levers or pulley systems are often used to provide the mechanical advantage necessary to increase the relatively slow motion of the pneumatic or hydraulic actuator to a high enough speed to facilitate flight for the UAV. The UAV is accelerated down a rail assembly by the motion generated by the actuator and is released from the launch platform at the end of the rail system. As with the hand launching method, the UAV's propeller(s) or propulsion system may either be operating prior to launch or be activated following its release from the launch platform. DOD UAVs that utilize pneumatic or hydraulic launch systems include the Navy's ScanEagle UAV [28], Army's RQ-7 Shadow [29], and the Army's Mk 4.7 Small Tactical UAV [30].

The use of a hydraulic or pneumatic powered launch system provides the user with several distinct advantages. First, no explosives or dedicated runways are required to operate these launch systems. This means the systems are generally safe and easy to transport and can be used in a wide range of locations and terrains. Additionally, since no collapsible wings or other mechanical flight surfaces are needed, the reliability and cost effectiveness of the aircraft are improved and, since they are not launched from tubes, the UAV configuration and payload options are expanded once again. Another benefit of these systems is that they can typically be reset quickly by simply reversing the direction of flow for the high pressure fluid. This rapid-reset capability could potentially be key to facilitating the large numbers of UAVs required to engage in swarm operations while minimizing the number of personnel required to execute the launches. Few systems currently in operation actually exploit this rapid-reset capability due to other launcher limitations. The primary disadvantage associated with pneumatic and hydraulic launch systems is the number of support systems that are required to operate the launcher. These systems require both a means of pressurizing the operating fluid, such as a compressor, and a means of storing this high pressure fluid prior to actuation of the system. These requirements result in a launch system that is both heavier than others and which requires some means of electrical power to operate; requirements that other launch systems do not have to account for. Finally, pneumatic and hydraulic powered launch systems must have some means of dissipating the high levels of heat and powerful forces generated during the launch of the aircraft, requiring a high degree of structural rigidity not associated with other launch systems. All these factors, taken together, explain why these launch systems are all fairly large in size, often requiring trucks or trailers to transport, as shown in Figure 2.6 and Figure 2.7.

2.1.5 Summary of Limitations

As alluded to previously, the primary downside of all these UAV launch systems is that they are either not capable of, or not designed to take advantage of a rapid reset following a UAV launch. While many UAVs can be launched quickly using the hand-launching method, there are safety, reliability, and personnel concerns that cannot be ignored. Conventional runway takeoffs are highly reliable, but require large flat areas, dedicated airstrips, and longer acceleration periods to accommodate launches which would make the rapid launch of many UAVs a highly complex evolution requiring careful coordination. Tube launchers are highly transportable and can be used in a wide variety of locations, but the safety

Figure 2.6: Pneumatic launch of Navy's ScanEagle UAV, from [28]

Figure 2.7: Pneumatic launch of Army's Mk 4.7 Small Tactical UAV, from [30]

concerns associated with the handing and transport of explosive materials and mechanically deployed flight surfaces limits their overall utility. Finally, pneumatically and hydraulically powered launchers facilitate high degrees of flexibility in terms of both UAV configuration and launch location and could be designed to quickly reset with little user input, but they also require many support systems and are generally difficult to transport and manipulate. A launch system which is expected to facilitate a swarm scenario using fixed-wing UAVs will ultimately need to incorporate some aspects of all these different systems in order to be truly effective. Rapid reset and launch capability, ease of transport, launch location flexibility, GCS integration, and user safety are all factors which would be highly desirable in the next-generation UAV launcher.

2.2 Scope of Proposed Solutions

For the purposes of this report, the design and development of a UAV launcher that is capable of rapidly launching a swarm of UAVs is focused towards the operation of a fleet of Ritewing Zephyr II UAVs [31]. This is the aircraft platform that NPS' ARSENL team is currently using in their research towards the implementation of UAV swarms. The Zephyr II UAV, shown in Figure 2.8, is essentially a styrofoam "flying wing" aircraft with a 52-inch wingspan which is propelled using a single, tail-based propeller and controlled using a pair of "elevons" [31]. The primary benefits provided by the Zephyr II aircraft are its relatively

low cost (approximately $1,200 for a complete aircraft build) and the ability to add additional sensors, processing, and communications systems by simply removing portions of styrofoam from carefully selected areas. These characteristics make the Ritewing Zephyr II an excellent platform for use in research and development applications in academic environments where concerns regarding cost and versatility are paramount. However, it is also worth noting that, due to the specific focus on the launch of the Zephyr II platform, some of the launch systems previously identified are sub-optimal solutions when considering the Zephyr's size and configuration constraints. For example, since the Zephyr II has no wheels or landing gear, a conventional runway takeoff would likely not be the best solution for getting this UAV airborne. Finally, due to this specific focus on the launch of a Zephyr II aircraft, it should be noted that launch solutions and specific decisions proposed through this work may not be directly applicable to other UAVs. However, the capabilities proposed and decision-making processes utilized herein are likely to have much more universal applicability.

Figure 2.8: A modified Ritewing Zephyr II UAV engaged in autonomous flight

Currently, the ARSENL team utilizes a bungee-operated launcher with a PVC rail system to get the Zephyr II aircraft airborne. The bungee cord is stretched and affixed to a hook located on the bottom of the UAV. The UAV is then held in place at the base of the PVC rail system by a launch technician while a second technician activates the onboard GoPro camera and performs the flight control surface checks. When ready, the launch technician receives final verification that the autopilot system is armed from the GCS operator and releases the aircraft, which is subsequently accelerated and elevated as the bungee contracts.

The UAV's propeller is then actuated by the onboard autopilot system once the aircraft detects a preset acceleration force followed quickly by a preset velocity value, indicating to the computer that the UAV has just experienced a launch. Eventually, the bungee contracts enough to release itself from the hook and the UAV commences normal powered flight.

Shown in Figure 2.9, this launch system is fairly reliable due largely to its simplicity, but it still has some important limitations. First, the bungee cord currently being used is nearly 100 feet in length when fully stretched. This requires one of the two launch technicians to walk nearly this entire distance following each launch to retrieve and re-stretch the cord when preparing for subsequent launches. Additionally, there are safety concerns inherent in this system since the launch technician holding the aircraft on the rails prior to launch is forced to sit only five to six inches from an armed propeller prior to launch. If a software problem in the onboard autopilot system were to unexpectedly actuate the UAV's motor prior to release, the launch technician holding the aircraft on the launch rails could potentially be injured by the spinning propeller. The system also requires extensive verbal communications between the launch technician and the GCS operator prior to launch, requires at least two technicians to operate, and is time-consuming to re-orient if wind direction or other environmental factors change. Ultimately, the ARSENL team needs to replace this launch system with one capable of higher launch rates, a high degree of integration with ground-based flight control systems, and a suite of sensor-based capabilities that have heretofore never been seen in a UAV launch system.

To accomplish these goals, a series of UAV launch system prototypes as described in this report were developed, tested, and iteratively improved upon. Each of these prototypes were expected to have significant mechanical design and implementation challenges, especially with regards to meeting stakeholder requirements for low weight, small footprint, minimal modification to the aircraft, and specific limits on velocity and acceleration profiles generated during launch. For reference, an in-depth overview of the mechanical design decision-making, development, and testing processes associated with the construction of these prototypes is provided by the author's partner in this effort, Raymond Davis, in his thesis entitled "Mechanical Design and Optimization of Swarm Capable UAV Launch Systems" [32]. However, while the optimal design of the primary mechanical systems associated with this new UAV launcher is certainly vital to ensuring its success, this report

Figure 2.9: Naval Postgraduate School's ARSENL team launching a Zephyr II UAV using the bungee-operated launch system

focuses primarily on the identification, selection, and development of the equally important software and sensors-based supporting capabilities that will enhance the launch system's utility, margin to safety, and improve the overall user experience.

These additional capabilities and system functions, while surely adding a degree of complexity to the system design, also help to facilitate faster, safer, more controlled launches while enabling new channels of communication between those involved in the launch process. Furthermore, it is reasonable to conceive that at least some of these sensors-based functions are absolutely critical to facilitating the proper and repeatable operation of some of the mechanical launching mechanisms. Finally, the identification, development, and implementation of each of these supporting capabilities are performed in parallel with the mechanical design and testing of each prototype launch system. Since mechanical designs may change drastically between iterations, it is vital that any solutions developed as part of

this work should be designed with platform-independence in mind, thereby enabling easy integration across a wide spectrum of mechanical design configurations.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:
Capability Identification and Prioritization

3.1 System Context

Before starting the process of identifying potential software and sensors-based capabilities that could be implemented into a new UAV launch system, it is prudent to understand the environment and context in which the system is expected to operate. According to the Defense Acquisition Guidebook (DAG):

> A system should not be acquired in isolation from other systems with which it associates in the operational environment. The Program Manager and Systems Engineer should understand how their system fills the needs for which it was designed and the enterprise context within which it operates. This includes understanding the diverse or dissimilar mix of other systems (hardware, software, and human) with which the system needs to exchange information. [33]

To this end, a context diagram depicting the adjustable and non-adjustable external systems with which the UAV launcher interacts is created. In conjunction with this process, all inputs and outputs from each of these external systems are also identified. The diagram, shown in Figure 3.1, clearly shows the launch system itself, all the external systems, and the information and materials that are expected to pass across each system boundary and interact with the launcher.

It is important to note in Figure 3.1 both the boundaries of the launch system itself and the information or materials flowing across these boundaries. The solid lines in the diagram represent information, materials, and efforts that are present with the existing bungee-operated launch system and should be accounted for in future system iterations. The dashed lines represent flow of information that does not currently exist, but which may be added to enable more efficient operation of the launch system. With this understanding of how the proposed launch system fits into its environment, the next priority is to identify and prioritize the enabling capabilities that facilitate the safe and efficient operation of a rapid

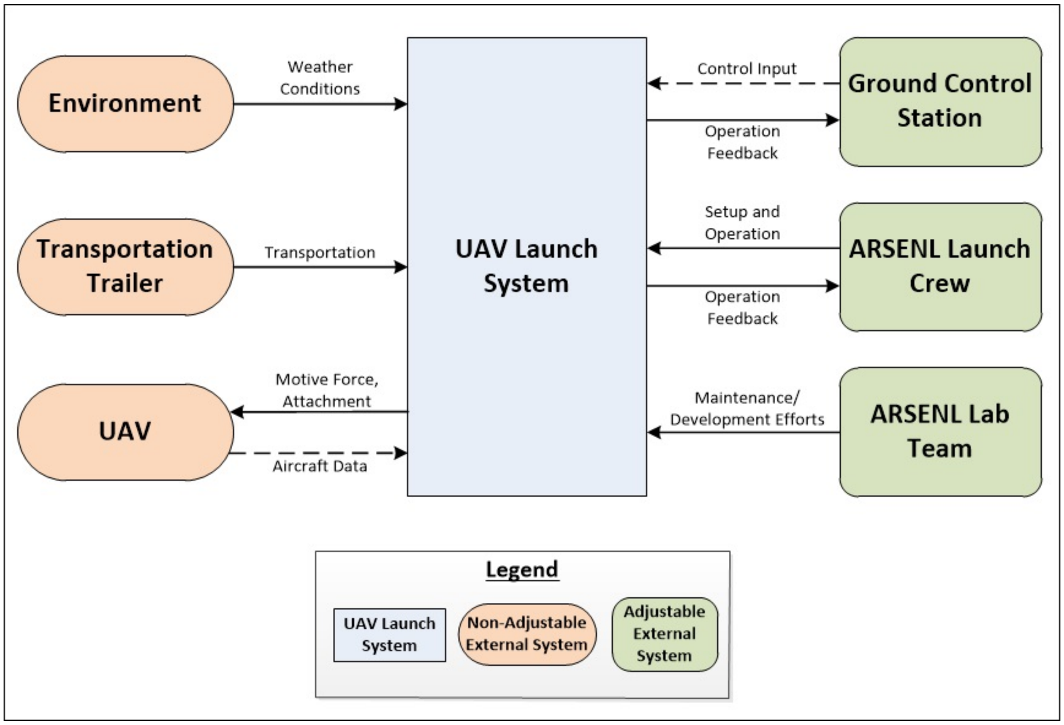

Figure 3.1: UAV launch system context diagram

UAV launch system.

3.2 Concept of Operations and Capability Identification

The JCIDS is a process used by DOD leadership to more efficiently identify and prioritize capability gaps in military operational and logistical applications and facilitate the development of procedural and materiel solutions to bridge these gaps [34]. The goal of the program is to "ensure a better understanding of the warfighting needs early in capability development and provide a more comprehensive set of valid, prioritized requirements," [9]. The "concept development" portion of the JCIDS process usually begins with a Capability Based Assessment (CBA) in which the overall mission and complete list of desired capabilities are identified [35]. Specific gaps in this list of capabilities are then highlighted, assessed, and prioritized so that recommendations regarding how best to work around or bridge these gaps can eventually be developed [35]. If a materiel solution is initially proposed, an Initial Capabilities Document (ICD) is generated to identify the Concept of Operations (CONOPS) for the employment of the solution system, any Joint Capability

Areas (JCAs) to which the system is expected to contribute, potential non-materiel alternatives to address the capability gap, and any recommendations for future-system requirements [35]. Once these capability documents and corresponding recommendations have been developed, system designers can use the system requirements identified during this initial portion of the JCIDS process as they begin designing systems to bridge the capability gap. Eventually, the various system design proposals will be evaluated against each other to identify the most cost-effective and efficient solution.

3.2.1 Operational Scenarios

Having already established the existence of a capability gap with regards to the DOD's ability to rapidly launch large numbers of UAVs and identified the need for a new materiel solution to this problem, the next prudent step is to identify a spanning set of operational scenarios in which the proposed launcher would be expected to operate. For the purposes of performing field launches of Zephyr II UAVs, three primary scenarios have been identified:

1. A single launcher performing launches of only one or two UAVs at a time
2. A single launcher performing consecutive launches of many UAVs
3. Two or more launchers performing consecutive launches of many UAVs

For each scenario, a concise summary of required capabilities are generated and applicable JCIDS JCAs are identified. However, since the primary goal of this launcher is to facilitate future research and development efforts for ARSENL team members, it should be understood that some portions of the JCIDS process, such as threat assessments and identification of capability overlaps, are necessarily neglected for this analysis since this particular UAV launcher is not expected to be utilized in a military operational environment.

Operational Scenario 1 – Single Launcher, Single UAV

This scenario is likely to play out at least once during nearly every one of the field exercise events where the ARSENL performs system developmental testing in support of its swarm UAV research. Prior to launch, the UAV is inspected and bench tested by ARSENL team members. This specifically includes the manual testing of all UAV control surfaces, extensive battery level and circuitry checks, propulsion motor response testing, and testing of all communications circuits expected to be in operation between the GCS and the UAV at both near and distant ranges. After completing these initial checks, the ARSENL team leader

obtains permission to execute a UAV launch and, once granted, a launch technician moves the UAV launch system into position and prepare it for launch.

The optimal position of the launch system in this scenario is dictated by several factors. First, the system needs to be located close enough to the GCS to ensure constant communications are maintained between the GCS and the launcher's onboard computer systems. The launcher also needs to be located in an area relatively clear of buildings and fixed obstacles, ground vehicles, other aircraft, and personnel foot traffic to minimize risks to personnel and equipment during and after launches. Weather conditions also play a role. An optimal UAV launch is directed into the wind in order to increase the aircraft's relative airspeed and, consequently, increase the lift forces generated over the airfoil. However, for takeoffs that take place with the wind directed behind the aircraft, the opposite effect occurs, reducing the lift force and increasing the probability of a stall during takeoff. Therefore, the launch system should be positioned and oriented, either manually or by automated means, to facilitate launches as closely as possible into the direction of wind.

After positioning, the launch technician needs to prepare the launcher and all its associated support systems for launch. Depending on the system's design and degree of complexity, this could include any number of steps such as energizing electrical systems, powering on computer systems, loading propellant charges or accelerants, pressurizing pneumatic or hydraulic tanks and hoses, establishing and verifying communications between launcher and GCS computer systems, and verifying the integrity of mechanical safety mechanisms and other interfaces. After completing this launcher set up, the launch technician ensures the launcher is reset and that any safety devices, if equipped, are engaged prior to loading the prepped and tested UAV onto the launcher's UAV interface. At this point, the technician further step clear of the immediate launch area, which upon being verified safe for launch by either the launch technician or independently by a launcher subsystem, transmits a signal to the GCS operator indicating the launcher's readiness. In conjunction with sending this readiness signal, the launcher should also activate some kind of warning system to notify personnel in the vicinity that the system is in a transient condition and a launch is imminent. When ready, the GCS operator remotely initiates the UAV launch and the launcher accelerate and release the UAV while dissipating any resultant forces generated from this process. During this acceleration and release process, the UAV's own automated

control systems turns on the power on the propeller motor once it reaches a preset velocity value. After launch, the technician or launch system re-verifies the launch area clear, and the system is reset either automatically or upon receipt of an input from the technician or GCS operator. Once reset, the system's safety mechanisms automatically re-engages, warning lights deactivate, and the launcher indicates to the technician that it is safe to load another aircraft. Since only one aircraft is being launched in this scenario, this concludes the launcher's expected set of operations.

A variety of capabilities need to be developed to support this operational scenario. First is the system's ability to be moved, maneuvered, set up and oriented by, at most, only one or two individuals. This means that the system needs to be relatively light, fairly compact, and any support system components should either be well-integrated or easily manipulated if separate from the main launching system. This system mobility functionality falls under the Force Application JCA, within the "Maneuver to Insert Air Assets" capability (*JCA 3.1.2.1*) [35]. The launcher also needs to be positioned in a location that supports reliable communications between its systems and the GCS. This requirement establishes range limitations that could vary based on which communication technologies are selected. These communications requirements fall under the Net-Centric JCA, within the "Wired" and "Wireless Transmission" capabilities (*JCAs 6.1.1 and 6.1.2*) [35].

Next, the launch system described in this scenario needs to be capable of detecting weather conditions in the immediate area with specific emphasis on wind speed and direction. Ideally, it is capable of disabling its ability to launch UAVs in the event that high speed crosswinds or tailwinds are detected and, while doing so, prompt the launch technician to reorient the system for a more favorable launch. An even more useful capability is the ability for the launcher to re-orient itself upon making such a determination, thereby removing the need for the technician to do so manually. Furthermore, the system needs to be able to detect unsafe conditions in the area immediately surrounding the launcher, such as obstacles in the flight path of the UAV or personnel standing too close to the system before or during operation. This includes verification that the launch technician is clear of the platform before activating the mechanical portions of the system for launch. These environmental detection capabilities fall under the Battlespace Awareness JCA, within the "Collect Atmospheric Environmental Measurements," "Analyze the Atmosheric and Land Environ-

ment," and "Assess Environmental Effects" capabilities (*JCAs 2.2.1.6, 2.2.2.1, 2.2.2.5, and 2.2.4.2*) [35]. The launcher used in this scenario should also be constructed, wired, and programmed in a manner that enables onboard electrical and computer systems to startup with minimal external guidance or user inputs. It should be able to frequently perform self-checks on electrical and mechanical systems to determine any anomalies and communicate any potential issues to either the launch technician or the GCS operator. These setup and self-monitoring capabilities fall under the Logistics JCA, within the "Maintenance Inspection," "Testing," and "Activation/Inactivation" capabilities (*JCAs 4.3.1, 4.3.2, and 4.3.3.1*) [35].

Once loaded, the launcher requires the capability to communicate its readiness for launch to the GCS operator. This feature informs the operator that a UAV is loaded, the system is reset and prepared for a launch, environmental conditions are favorable, and that the launch technician is adequately clear of the platform. This is also where the ability to warn personnel in the area that a system actuation is imminent is useful, likely in the form of some kind of highly visible lighting system. Following launch and reset, this same lighting system also can give the launch technician some kind of "safe to load" indication. Finally, while not specifically mentioned in the above scenario, it is also prudent to include some means for the launch technician to deactivate the system at any point up to and during launch execution. By including this "Abort" switch within easy reach of the technician, the possibility of a launch occurring when conditions are changing too rapidly for the automated portions of the launch system to detect can be reduced and, in doing so, risks to both the UAVs and personnel working in the area is minimized. Furthermore, functionality could potentially be added that would enable either the GCS operator, safety observer, or mission commander to halt the launch process if any of these parties detects an abnormal or unsafe condition. All these functions likely fall under the Command and Control JCA, within the "Share Knowledge and Situational Awareness," "Manage Risk," and "Provide Warnings" capabilities (*JCAs 5.2.3, 5.4.1, and 5.5.1.5*) [35].

Operational Scenario 2 – Single Launcher, Multiple UAVs

The second operational scenario plays out slightly less frequently than the first, but is anticipated to still be quite common as the ARSENL continues to develop the swarm-driven technological capabilities. As with the first scenario, it is assumed that any UAVs involved

in a multiple-unit launch event have all mechanical, electrical, and communications systems inspected or bench tested by team members prior to commencing operations. Once the checks are complete on all aircraft planned for launch, the ARSENL team leader obtains permission to execute multiple UAV launches and the launch technician begins moving the launcher and any subsystems into position and preparing them for operation. As before, the launch system needs to be located close enough to the GCS to ensure the maintenance of constant communications between the systems, but also is required to remain sufficiently clear of buildings, ground vehicles, other aircraft, and personnel foot traffic. Additionally, the environmental conditions previously discussed (e.g., wind speed and direction) continue to play a significant role in facilitating the execution of a multiple-unit launch event.

After positioning and orienting the system, the launch technician prepares the launcher and all its associated support systems for operation by energizing electrical systems, powering on computer systems, loading propellant charges or accelerants, pressurizing pneumatic or hydraulic tanks and hoses, establishing and verifying communications between launcher and GCS computer systems, and verifying the integrity of mechanical safety mechanisms and other interfaces. The technician further ensures the launcher is reset, safety devices are engaged, and commences loading the first UAV onto the launcher's UAV interface. Next, the technician steps clear of the launch area and, once verified safe, the system transmits a signal to the GCS operator indicating its readiness to launch while activating the onboard warning system to notify personnel in the vicinity that a launch is imminent. Again, the UAV launch is remotely initiated by the GCS operator, and the launcher accelerates and releases the aircraft while dissipating the forces generated in the process. The area is verified clear by either the technician or the launch system itself, and the launcher is reset either automatically or upon receipt of an electrical or mechanical input from the technician or GCS operator. Once reset, the system's safety mechanisms automatically re-engage, the warning system deactivated, and a visual signal indicating that the system is in a safe condition is illuminated. At this point, the technician should already be holding and ready to load the next UAV. With the safety signal illuminated, the launch technician loads this next aircraft onto the launcher's interface and again steps clear. At this point, the safety verification, launch, reset, and UAV loading processes is repeated in short succession for as many aircraft as are required to be launched.

As the majority of this second scenario tracks closely with the first, all the enabling capabilities identified in the previous section also apply here. However, due to the need to quickly launch multiple UAVs for this scenario, some of the capabilities identified become significantly more important. Specifically, the automation of many of the launcher's mechanical functions enables the technician to pay less attention to the specific operation of the launcher and instead focus on the retrieval and loading of UAVs as quickly as possible. To facilitate this automation, the ability for the launcher to detect unfavorable wind or other environmental conditions, to include the presence of personnel in the immediate vicinity, becomes all the more important. However, the launcher must now be able to recognize these conditions, interpret them, and then take appropriate action with as little input from the technician as possible. This means that providing the launcher with the capability to re-orient itself to optimize launch conditions becomes more important as well since winds could easily shift or change during the time UAVs are being launched. As before, the ability to detect and interpret these environmental factors falls largely under the Battlespace Awareness JCA (*JCAs 2.2.1.6 , 2.2.2.1, 2.2.2.5, and 2.2.4.2*), but this time would include aspects of the Command and Control capability area as well (*JCAs 5.2.3, 5.4.1, 5.4.2.1, and 5.5.1.5*) [35].

With automated functionality and, if the launch system is able to detect the presence of an aircraft loaded on its UAV interface or launch platform, the operator at the GCS could benefit from increased situational awareness regarding the launcher's status. This awareness is also enhanced if the launcher is embedded with platform position sensors that would inform the GCS of whether the platform has been reset and is awaiting a UAV, launched and awaiting reset, or in a transient state between the two conditions. Furthermore, if an organization is conducting swarm operations with a set of UAVs that each had different functional capabilities, a useful feature of the launcher would be for the GCS operator or Swarm Commander to know not only when a UAV had been loaded for launch, but also that UAV's specific type or unit number. This information could help operators time launches more effectively based on the specific capabilities of the aircraft loaded on the platform and the understanding of the capabilities required at that point in the mission. These functions fall under both the Logistics and Command and Control JCAs, within the "Sustain the Force," "Share Knowledge and Situational Awareness," and "Select Course of Action" capabilities (*JCAs 4.1.2, 5.2.3, and 5.4.2.1*) [35].

Operational Scenario 3 – Multiple Launchers, Multiple UAVs

The third and final operational scenario outlines an event sequence that is highly probable if swarm UAV operations are ever adapted to a military operational context. This scenario would be executed when ARSENL reaches a point where they are ready to test swarm capabilities with very large numbers of aircraft. As with the previous two scenarios, it is again assumed that any UAVs involved in this multiple-unit launch event has all mechanical, electrical, and communications systems inspected or bench tested by team members prior to commencing launching. Once all checks are complete on all aircraft, the team leader obtains permission to commence launches in support of swarm UAV operations and the launch technicians begin moving two or more launchers, along with any subsystems, into position. All launch systems again need to be located close enough to the GCS to ensure the ability to maintain communications, while remaining sufficiently clear of buildings, ground vehicles, other aircraft, and personnel foot traffic. Launcher positioning and orientation in this scenario, however, are further restricted by the positions and orientations of the other launch platforms being used by the organization. Each launcher needs to be positioned sufficiently clear of the others to ensure that technicians are not inside any sister launchers' operating envelopes when traveling to or loading a launch system for operation. Finally, just as before, environmental conditions such as wind direction and speed continue to play a role in determining the optimal orientation for each launch system involved in the operation.

After positioning and orienting the system, the launch technician again prepares the launcher by energizing electrical systems, powering on computer systems, pressurizing pneumatic or hydraulic tanks and hoses, establishing and verifying communications between launcher and GCS computer systems, and verifying the integrity of mechanical safety mechanisms and other interfaces. The technician then ensures the launcher is reset with safety devices engaged before loading the first UAV onto the launcher's UAV interface. Next, the technician steps away from the immediate launch area and, once verified safe, the system transmits a signal to the GCS operator indicating its readiness to launch. At the same time, the launcher activates the onboard warning system to notify personnel in the vicinity of an imminent launch. As in the previous two scenarios, the UAV launch is remotely initiated by the GCS operator, and the launcher accelerates and releases the aircraft while dissipating any forces generated. The launch area is once again verified clear

by either the technician or some launcher subsystem, and the launcher is reset either automatically or upon receipt of an input from the technician or GCS operator. Once reset, the system's safety mechanisms automatically re-engage, the warning system is deactivated, and a visual signal indicating that the system is in a safe condition is illuminated to inform the technician that he may safely load the next aircraft. When the safe-to-load signal is illuminated, the launch technician loads the next UAV onto the launcher's interface and again step clear. The safety verification, launch, reset, and UAV loading processes is then rapidly repeated for as many aircraft as are required to be launched.

Once again, many of the capabilities required in this third launch scenario directly correspond to those previously identified. As such, it is immediately recognized that all the enabling capabilities identified in both the first and second scenarios would apply here and, as in the second scenario, the ability to maximize launcher automation plays a key role in enabling fast, safe, and efficient UAV launches. Enabling technologies specifically include the launchers' ability to detect unfavorable wind conditions, the presence of personnel in the immediate vicinity, and the ability to detect and or identify UAVs loaded onto the system. However, this third scenario introduces a unique dynamic not previously accounted for as new launch platforms are introduced to the launch event. The use of these multiple launch systems creates the need for several new capabilities. First, the use of two launch systems to get large numbers of UAVs airborne significantly increases the importance of proper platform positioning and orientation. To better facilitate this, launch systems may be able to identify the positions, orientations, and launch statuses of other systems operating nearby. For example, if Launcher A were able to detect the position and orientation of Launcher B, it might be able to disable its own launch capability and alert the GCS operator and launch technicians that Launcher B is either located too close for safe operation or that it is oriented in a direction that would place personnel and equipment near Launcher B in danger. Additionally, Launcher B should be able to make these same determinations and should disable its own launch capabilities if these events were to occur. These abilities are even more critical if the launchers are both equipped with enabling technologies that facilitated automated re-orientation to account for changing environmental conditions. All these inter-system communications and decision-making capabilities fall under the Net-Centric and Command and Control JCAs, specifically within the "Wired" and "Wireless Transmission," "Share Knowledge and Situational Awareness," and "Select Course of Action"

capabilities (*JCAs 6.1.1, 6.1.2, 5.2.3, and 5.4.2.1*) [35].

In a related situation, if both launchers were to launch UAVs at the same instant, both the GCS operator and his supporting computer systems might be temporarily overloaded with information. This event could cause unnecessary confusion for the operator and, in limited bandwidth situations, could cause significant buffering issues for GCS computers as large volumes of rapidly changing information is transmitted and processed. To prevent such an event, launchers capable of communicating their system statuses with each other as well as the GCS is beneficial. Then, if one launcher detects that the other is ready to launch, it could disable its own launch capabilities until the other launcher's cycle is complete. These abilities again fall under the Net-Centric and Command and Control JCAs, within the "Wired and Wireless Transmission," "Share Knowledge and Situational Awareness," "Manage Risk," and "Select Course of Action" capabilities (*JCAs 6.1.1, 6.1.2, 5.2.3, 5.4.1, and 5.4.2.1*) [35].

3.3 Summary of Potential Capabilities

For ease of reference, Table 3.1 summarizes all the capabilities that have been proposed through the development of these three Operational Scenarios and lists the corresponding JCAs.

Table 3.1: Potential capabilities summary with JCAs, after [35]

Capability	**Tier 1 JCA(s)**		**Specific JCA(s)**	
System moved and setup by 1-2 technicians	3.0	Force Application	3.1.2.1	Maneuver to Insert Air Assets
Streamline setup and initialization	4.0	Logistics	4.3.1 4.3.2 4.3.3.1	Inspect Test Activate/Inactivate
Detect environmental parameters *(wind data)*	2.0	Battlespace Awareness	2.2.1.6 2.2.2.5 2.2.4.2	Collect Atmospheric Measurements Analyze Atmospheric Environment Assess Environmental Effects
Re-orient launcher if wind direction not favorable	5.0	Command & Control	5.2 5.4.1 5.4.2	Understand Manage Risk Select Actions

continued ...

. . . Table 3.1 continued

Capability	Tier 1 JCA(s)		Specific JCA(s)	
Disable launch capability if winds averse	5.0	Command & Control	5.2 5.4.1 5.4.2 5.5.1.5	Understand Manage Risk Select Actions Provide Warnings
Maximize launcher range envelope from GCS	6.0	Net-Centric	6.1.1 6.1.2	Wired Transmission Wireless Transmission
Detect people/objects in launcher vicinity	2.0	Battlespace Awareness	2.2.1.1 2.2.2.1 2.2.4.2	Collect Land Measurements Analyze Land Environment Assess Environmental Effects
Disable launch ability if launch area unsafe	5.0	Command & Control	5.2 5.4.1 5.4.2 5.5.1.5	Understand Manage Risk Select Actions Provide Warnings
Automatic reset capability	4.0	Logistics	4.1.2	Sustain the Force
Lighting system to warn personnel of launch status	5.0	Command & Control	5.2 5.4.1 5.5.1.5	Understand Manage Risk Provide Warnings
Safe to load indication	5.0	Command & Control	5.2 5.4.1 5.5.1.5	Understand Manage Risk Provide Warnings
Abort launch functionality	5.0	Command & Control	5.4.1	Manage Risk
Mechanical-based kill switches with easy accessibility	5.0	Command & Control	5.4.1	Manage Risk
Communicate launch system/sensor status to GCS	5.0	Command & Control	5.4.1 5.5.1.5	Manage Risk Provide Warnings
Receive "Halt Launch" commands from safety observers	5.0	Command & Control	5.4.1 5.5.1.5	Manage Risk Provide Warnings
Detect UAV on launch platform	5.0	Command & Control	5.2 5.2.3	Understand Share Knowledge/Situation Awareness
Disable launch ability until UAV loaded	5.0	Command & Control	5.4.1 5.4.2	Manage Risk Select Actions
Identify UAV on launch platform	4.0 5.0	Logistics Command & Control	4.1.2 5.2.3	Sustain the Force Share Knowledge/Situation Awareness

continued . . .

. . . Table 3.1 continued

Capability	Tier 1 JCA(s)		Specific JCA(s)	
Launch platform position sensors	5.0	Command & Control	5.2.3	Share knowledge/Situation Awareness
Detect other launch system position/orientation	6.0	Net-Centric	6.1.1 6.1.2	Wired Transmission Wireless Transmission
Disable launch ability if oriented towards other launcher	5.0	Command & Control	5.4.1 5.4.2	Manage Risk Select Actions
Alert technician of competing orientation problems	5.0	Command & Control	5.2 5.5.1.5	Understand Provide Warnings
Detect other launch system's launch status	6.0	Net-Centric	6.1.1 6.1.2	Wired Transmission Wireless Transmission
Disable if other launch system is in launch cycle	5.0	Command & Control	5.2 5.4.1 5.4.2	Understand Manage Risk Select Actions

3.4 Capability Prioritization

The list of potential capabilities identified in Table 3.1 is robust enough that it would be foolhardy to attempt the development of all capabilities at one time. Instead, it is much more efficient to identify limited subsets of these capabilities and then work to implement each subset in an iterative prototyping process. This enables the system designer to better focus his efforts and, when problems arise at the implementation stage of the process, troubleshoot these issues on a more limited scale. The question then becomes how best to prioritize and group these capabilities into more manageable subsets?

First, it is logical that those capabilities that contribute to all three operational scenarios should be considered more strongly than those that only contribute to one or two. Similarly, capabilities contributing to two scenarios should be more heavily weighted than those only contributing to one. This helps ensure that those capabilities that will be utilized the most, that is, those used in more launching scenarios, will be developed first. Next, it is useful to rank each of the capabilities based on their general contributions to the overall launch system. For example, some of the capabilities listed, such as mechanical based kill switches with easy accessibility or launch platform position sensors, contribute directly to the system's ability to launch UAVs effectively. Such capabilities are therefore categorized

as “launch critical.” Others, such as detecting environmental characteristics or detecting the presence of a UAV on the launch platform, also contribute to launch, but in a less direct manner. Instead, these capabilities are geared more towards the optimization and automation of the launch process. Still other capabilities contribute by only enhancing the user interface with the launcher. This makes the launch process easier for the launch technician, but does not necessarily make a great difference in the ability to take advantage of the rapid-launch functionality provided by the system. Therefore, it follows that those capabilities that contribute directly and are deemed critical to the launch process receive the highest weight in the decision-making process, while those that only indirectly impact the launching process receive the lowest weights.

Finally, the anticipated difficulty associated with implementing each capability should be considered, with those that are easiest to implement receiving more weight than those that are more difficult. This helps to ensure that easier capabilities are developed first, providing a foundation to build upon as more difficult challenges are attacked in subsequent prototype iterations. Furthermore, this metric provides a means of capturing potential schedule risk inherent in the capability development process; if a capability is difficult to implement, it will likely take longer to complete and integrate with the rest of the system. Conversely, easier capabilities are likely to be implemented and integrated much more quickly, presenting less risk to the overall system development schedule. It should be noted, however, that while useful for capability prioritization, this “expected difficulty” metric is the most subjective of all the factors included in this decision-making process since it is based largely on the designer’s own biases, previous experience, and expectations.

One factor that is usually key to decision-making processes has been excluded from this analysis: budgetary concerns. For the purposes of developing this new launch system, the project sponsor and stakeholders placed a higher priority on acquiring the launch system than on cost and, as such, no specific budget limits are established. Instead, as development progressed, estimated costs and benefits are reviewed with the sponsor and funding is approved on a rolling basis. Thus, the capability implementation costs are not determined to be a critical factor for making prioritization decisions and are therefore excluded from this decision analysis.

Having identified these decision-making metrics, the initial capability scoring matrix

shown in Table 3.2 is created. All the possible capabilities identified in the previous section are listed and the operational scenarios, capability utility category, and expected degree of difficulty associated with implementing each capability are identified. As discussed previously, those capabilities that contribute to more scenarios, provide more important functionality to the launch system, or are easier to implement receive the highest scores, while those that contribute to only one scenario or are expected to be difficult to implement receive the lowest scores. For ease of identification later in this process, those capabilities that are directly dependent on the successful implementation of a different capability in the table are shown in bold, italic font. Ultimately, this initial scoring matrix provides a useful starting point for beginning an Analytic Hierarchy Process (AHP) analysis for multivariate decision-making, which helps prioritize the most important capabilities for development.

Table 3.2: Initial capability scoring matrix (Items in ***bold italics*** indicate dependence on other capabilities)

Capability	*Scenario Contribution*		*Utility Provided to User*		*Expected Implementation Difficulty*	
	# Scenarios Contributed	**Scenarios Score**	**Utility Provided**	**Nominal Utility Score**	**Expected Difficulty**	**Nominal Difficulty Score**
Moved and setup by 1-2 technicians	1 2 3	3	Launch-Critical	3	Easy	3
Streamline setup and initialization	1 2 3	3	Optimizes Launch Process	2	Medium	2
Detect environmental parameters (wind data)	1 2 3	3	Optimizes Launch Process	2	Easy	3
Re-orient launcher if wind direction not favorable	1 2 3	3	Enhances System Interfaces	1	Hard	1
Disable launch capability if winds adverse	1 2 3	3	Enhances System Interfaces	1	Easy	3
Maximize launcher range envelope from GCS	1 2 3	3	Optimizes Launch Process	2	Medium	2
Detect people/objects in launcher vicinity	1 2 3	3	Optimizes Launch Process	2	Medium	2
Disable launch ability if launch area unsafe	1 2 3	3	Enhances System Interfaces	1	Medium	2
Automatic reset capability	1 2 3	3	Optimizes Launch Process	2	Medium	2
Lighting system to warn personnel of launch status	1 2 3	3	Launch-Critical	3	Easy	3
Safe to load indication	1 2 3	3	Enhances System Interfaces	1	Easy	3
Abort launch functionality	1 2 3	3	Launch-Critical	3	Easy	3
Mechanical-based kill switches with easy accessibility	1 2 3	3	Launch-Critical	3	Easy	3
Communicate launch system/sensor status to GCS	1 2 3	3	Enhances System Interfaces	1	Hard	1
Receive 'Halt Launch' command from safety observers	1 2 3	3	Enhances System Interfaces	1	Hard	1
Detect UAV on launch platform	2 3	2	Optimizes Launch Process	2	Medium	2
Disable launch ability until UAV loaded	2 3	2	Enhances System Interfaces	1	Medium	2
Identify UAV on launch platform	2 3	2	Enhances System Interfaces	1	Medium	2
Launch platform position sensors	2 3	2	Launch-Critical	3	Easy	3
Detect other launch system position/orientation	3	1	Enhances System Interfaces	1	Hard	1
Disable launch ability if oriented towards other launcher	3	1	Enhances System Interfaces	1	Medium	2
Alert technician of competing orentation problems	3	1	Enhances System Interfaces	1	Easy	3
Detect other launch system's launch status	3	1	Enhances System Interfaces	1	Hard	1
Disable if other launch system is in launch cycle	3	1	Enhances System Interfaces	1	Easy	3

The AHP is a structured approach for determining the scores and weights used in a multi-criteria scoring model when the decision maker finds it difficult to define them subjectively [36]. This process is particularly useful when trying to use multiple criteria to make useful, logical decisions pertaining to a large number of alternatives [36]. The use of an AHP approach for prioritizing the launch system capabilities previously identified is logical due to the large number of alternatives (the capabilities), the use of multiple decision criteria (# Scenarios, Utility Provided, and Expected Difficulty), and the difficulty inherent in comparing the benefits provided by each of these criteria to each alternative in an academically rigorous and logically sound manner.

For example, in Table 3.2, the "Re-orient launcher if wind direction not favorable" capability received a nominal Scenario Score of three, but received a value of one for the Nominal Utility and Nominal Difficulty Scores since it was expected to be difficult to implement and only represented an enhancement of the launch system interface. How then can one objectively say whether the value provided by a contribution to all three scenarios is important enough to outweigh the lower utility and difficulty measures? Furthermore, for each criterion in the table, only three possible values were assigned, essentially representing a "Good-Better-Best" valuation of each alternative under that criterion. However, a score of two for an alternative under a given criterion may not necessarily be twice as good as a score of one and, similarly, a score of three may not be 50% better than a score of two or three times better than a score of one. Ultimately, the AHP method helps resolve all these issues and also provides a method by which one can verify the consistency of the prioritization decisions made for a given criterion [36].

The AHP begins with the creation of a pairwise comparison matrix for each decision-making criterion using all possible alternatives [36]. In this matrix, individual decisions are made by prioritizing the relative importance of each alternative compared to each of the other alternatives while taking into account only a single decision criterion [36]. This process is independently repeated for each of the other decision-making criteria, and then is also used to determine the weighting values that correspond to each of those decision criteria [36]. An example pairwise comparison matrix for the Expected Difficulty criterion is shown in Table 3.3.

Table 3.3: Pairwise comparison matrix for the Expected Difficulty criterion

	Moved and setup by 1-2 technicians	Streamline setup and initialization	Detect environmental parameters (wind data)	Re-orient launcher if wind direction not favorable	Disable launch capability if winds adverse	Maximize launcher range envelope from GCS	Detect people/objects in launcher vicinity	Disable launch ability if launch area unsafe	Automatic reset capability	Lighting system to warn personnel of launch status	Safe to load indication	Abort launch functionality	Mechanical-based kill switches with easy accessibility	Communicate launch system/sensor status to GCS	Receive 'Halt Launch' command from safety observers	Detect UAV on launch platform	Disable launch ability until UAV loaded	Identify UAV on launch platform	Launch platform position sensors	Detect other launch system position/orientation	Disable launch ability if oriented towards other launcher	Alert technician of competing orentation problems	Detect other launch system's launch status	Disable if other launch system is in launch cycle
Moved and setup by 1-2 technicians	1	2	1	3	1	2	2	2	2	1	1	1	1	3	3	2	2	2	1	3	2	1	3	1
Streamline setup and initialization	0.5	1	0.5	2	0.5	1	1	1	1	0.5	0.5	0.5	0.5	2	2	1	1	1	0.5	2	1	0.5	2	0.5
Detect environmental parameters (wind data)	1	2	1	3	1	2	2	2	2	1	1	1	1	3	3	2	2	2	1	3	2	1	3	1
Re-orient launcher if wind direction not favorable	0.33	0.5	0.33	1	0.33	0.5	0.5	0.5	0.5	0.33	0.33	0.33	0.33	1	1	0.5	0.5	0.5	0.33	1	0.5	0.33	1	0.33
Disable launch capability if winds adverse	1	2	1	3.00	1	2	2	2	2	1	1	1	1	3	3	2	2	2	1	3	2	1	3	1
Maximize launcher range envelope from GCS	0.5	1	0.5	2	0.5	1	1	1	1	0.5	0.5	0.5	0.5	2	2	1	1	1	0.5	2	1	0.5	2	0.5
Detect people/objects in launcher vicinity	0.5	1	0.5	2	0.5	1	1	1	1	0.5	0.5	0.5	0.5	2	2	1	1	1	0.5	2	1	0.5	2	0.5
Disable launch ability if launch area unsafe	0.5	1	0.5	2	0.5	1	1	1	1	0.5	0.5	0.5	0.5	2	2	1	1	1	0.5	2	1	0.5	2	0.5
Automatic reset capability	0.5	1	0.5	2	0.5	1	1	1	1	0.5	0.5	0.5	0.5	2	2	1	1	1	0.5	2	1	0.5	2	0.5
Lighting system to warn personnel of launch status	1	2	1	3.00	1	2	2	2	2	1	1	1	1	3	3	2	2	2	1	3	2	1	3	1
Safe to load indication	1	2	1	3.00	1	2	2	2	2	1	1	1	1	3	3	2	2	2	1	3	2	1	3	1
Abort launch functionality	1	2	1	3.00	1	2	2	2	2	1	1	1	1	3	3	2	2	2	1	3	2	1	3	1
Mechanical-based kill switches with easy accessibility	1	2	1	3.00	1	2	2	2	2	1	1	1	1	3	3	2	2	2	1	3	2	1	3	1
Communicate launch system/sensor status to GCS	0.33	0.5	0.33	1	0.33	0.5	0.5	0.5	0.5	0.33	0.33	0.33	0.33	1	1	0.5	0.5	0.5	0.33	1	0.5	0.33	1	0.33
Receive 'Halt Launch' command from safety observers	0.33	0.5	0.33	1	0.33	0.5	0.5	0.5	0.5	0.33	0.33	0.33	0.33	1	1	0.5	0.5	0.5	0.33	1	0.5	0.33	1	0.33
Detect UAV on launch platform	0.5	1	0.5	2	0.5	1	1	1	1	0.5	0.5	0.5	0.5	2	2	1	1	1	0.5	2	1	0.5	2	0.5
Disable launch ability until UAV loaded	0.5	1	0.5	2	0.5	1	1	1	1	0.5	0.5	0.5	0.5	2	2	1	1	1	0.5	2	1	0.5	2	0.5
Identify UAV on launch platform	0.5	1	0.5	2	0.5	1	1	1	1	0.5	0.5	0.5	0.5	2	2	1	1	1	0.5	2	1	0.5	2	0.5
Launch platform position sensors	1	2	1	3.00	1	2	2	2	2	1	1	1	1	3.00	3.00	2	2	2	1	3	2	1	3	1
Detect other launch system position/orientation	0.33	0.5	0.33	1	0.33	0.5	0.5	0.5	0.5	0.33	0.33	0.33	0.33	1	1	0.5	0.5	0.5	0.33	1	0.5	0.33	1	0.33
Disable launch ability if oriented towards other launcher	0.5	1	0.5	2	0.5	1	1	1	1	0.5	0.5	0.5	0.5	2	2	1	1	1	0.5	2	1	0.5	2	0.5
Alert technician of competing orentation problems	1	2	1	3.00	1	2	2	2	2	1	1	1	1	3.00	3.00	2	2	2	1	3.00	2	1	3	1
Detect other launch system's launch status	0.33	0.5	0.33	1	0.33	0.5	0.5	0.5	0.5	0.33	0.33	0.33	0.33	1	1	0.5	0.5	0.5	0.33	1	0.5	0.33	1	0.33
Disable if other launch system is in launch cycle	1	2	1	3.00	1	2	2	2	2	1	1	1	1	3.00	3.00	2	2	2	1	3.00	2	1	3.00	1
Column Sum	16.2	31.5	16.2	53.0	16.2	31.5	31.5	31.5	31.5	16.2	16.2	16.2	16.2	53.0	53.0	31.5	31.5	31.5	16.2	53.0	31.5	16.2	53.0	16.2

In this table, each of the capability alternatives appear on both the row and column headers. A value was then assigned to each intersection to annotate the degree to which the alternative listed on the row was preferred to the alternative listed on the column corresponding to that intersection. A value of one indicates a situation where neither alternative is preferred over the other. A value of two indicates that the alternative listed on the row is moderately preferred over the alternative listed in the corresponding column. Finally, a value of three indicates that the alternative listed on the corresponding row is strongly preferred over the alternative listed on the intersecting column. Similarly, a value of 0.5 indicates that the value listed in the corresponding column is moderately preferred over the alternative shown in the intersecting row, and a value of 0.33 indicates a strong preference for the alternative in the column.

For example, in the Initial Capability Scoring Matrix shown previously in Table 3.2, the "Moved and setup by one to two technicians" capability was assigned an Expected Difficulty of "Easy," with a corresponding Nominal Difficulty Score of three points. The second capability in that matrix, "Streamline setup and initialization," was expected to be slightly harder to implement, with an Expected Difficulty of "Medium" and a corresponding Difficulty Score of two points. Now, moving down to the Pairwise Comparison Matrix (Table 3.3), first note the top row, which corresponds to the "Moved and setup by one to two technicians" capability. Moving to the right across this row, it is apparent that the first column in the comparison table also corresponds to this same capability and, since a nominal value of three is being compared against against the same nominal value of three, a comparison score of one was assigned at this intersection. As one would expect, this indicates that there is no significant preference when comparing the "Moved and setup by one to two technicians" capability against itself. Continuing to the right in this same first row, the second column corresponds to the "Streamline setup and initialization" capability. Since this capability had been assigned a Nominal Difficulty Score of two in Table 3.2, a comparison score of two was assigned at this intersection. This indicates that the "Moved and setup by one to two technicians" capability is moderately preferred over the "Streamline setup and initialization" capability when considering only the Expected Implementation Difficulty criterion. This same process was then repeated for all remaining capability intersections in the table and was subsequently repeated in full for both the # Scenarios and Utility Provided criteria. This resulted in a set of three pairwise comparison matrices that annotated

individual capability preference decisions based on each of the decision-making criteria.

The next step in the AHP process is to normalize the values in each of the pairwise comparison matrices [36]. To do this, a new, normalized pairwise comparison matrix is generated by dividing the value in each individual cell by the sum of all the values in its corresponding column. When complete, an average of the normalized values in each row is calculated, producing what amounts to a normalized, unweighted score for each capability alternative under the specific decision criterion.

At this point, the decision criteria weighting values must be determined in order to scale these normalized scores and tabulate final scores for each alternative [36]. To generate these values, a fourth set of pairwise comparison matrices was created in which the three decision criteria are prioritized against one another. For these criteria weighting comparison tables, which are shown in Table 3.4 and Table 3.5, it was decided that the most important criterion is Utility. The reasoning here was that, regardless of the ease with which a capability can be implemented or the number of scenarios to which it is expected to contribute, one would always prefer to develop those capabilities that are classified as "Launch Critical" first, thereby facilitating a baseline level of launch system operation. The second-most important criterion was determined to be the number of scenarios to which a capability is expected to contribute since the Expected Difficulty criterion, while important, is still a highly subjective measure based largely on assumptions rather than real data or information. Thus, for the purposes of generating the weight values associated with the decision criteria, Utility was annotated as Moderately Preferred over the # Scenarios criterion, and as Strongly Preferred over the Expected Difficulty criterion.

Table 3.4: Piecewise comparison matrix for determining criteria weighting values

	# Scenarios	Utililty	Expected Difficulty
# Scenarios	1	0.5	2
Utility	2	1	3
Expected Difficulty	0.5	0.333	1
Column Sum	3.5	1.833	6

Table 3.5: Normalized piecewise comparison matrix for criteria weights with final weighting scores

	# Scenarios	Utililty	Expected Difficulty	***Weighting Score***
# Scenarios	0.286	0.273	0.333	***0.297***
Utility	0.571	0.545	0.5	***0.539***
Expected Difficulty	0.143	0.182	0.167	***0.164***

As with the other piecewise comparison matrices, the values within the Criteria Weights Piecewise Comparison matrix were normalized and then averaged across each criterion, generating a column of corresponding swing weight values. These decision-weighting values are 0.539 for the Utility Provided metric, 0.297 for the # Scenarios metric, and 0.164 for the Expected Difficulty metric.

At this point, all the necessary information was available to facilitate generation of the final composite prioritization scores for each capability. To obtain these scores, each criterion weight value was first multiplied by all the normalized, unweighted scores generated from the piecewise comparison in which that same criterion was the primary metric. This produced a set of three weighted scores for each capability alternative that correspond to the three decision criteria. Adding these weighted scores together produced the final composite prioritization scores for each capability. For this analysis, higher scores designate more desirable capabilities, while lower scores correspond to the less important capabilities.

The complete, prioritized capability list generated by this AHP is shown in Table 3.6. Once again, for clarity and better understanding, those capabilities that are dependent on the development of another are italicized and highlighted in bold. It is recommended that these functions be reserved for later prototype iterations after the corresponding initial capabilities have been implemented.

Table 3.6: Nominal AHP scores and ranking for all potential capabilities (Items in ***bold italics*** indicate dependence on other capabilities)

Capability	AHP Score
Abort launch functionality	0.0689
Mechanical-based kill switches with easy accessibility	0.0689
Lighting system to warn personnel of launch status	0.0689
Moved and setup by 1-2 technicians	0.0689
Launch platform position sensors	0.0615
Detect environmental parameters (wind data)	0.0511
Streamline setup and initialization	0.0464
Maximize launcher range envelope from GCS	0.0464
Detect people/objects in launcher vicinity	0.0464
Automatic reset capability	0.0464
Safe to load indication	0.0393
Disable launch capability if winds adverse	0.0393
Detect UAV on launch platform	0.0390
Disable launch ability if launch area unsafe	0.0345
Communicate launch system/sensor status to GCS	0.0322
Receive 'Halt Launch' command from safety observers	0.0322
Re-orient launcher if wind direction not favorable	0.0322
Disable if other launch system is in launch cycle	0.0284
Alert technician of competing orentation problems	0.0284
Disable launch ability until UAV loaded	0.0271
Identify UAV on launch platform	0.0271
Disable launch ability if oriented towards other launcher	0.0237
Detect other launch system's launch status	0.0214
Detect other launch system position/orientation	0.0214

3.5 Sensitivity Analysis

As this AHP decision-making analysis is highly dependent on the largely subjective assignment of utility and difficulty scores for each of the potential capabilities, it is useful to perform a sensitivity analysis on these subjective criteria to better understand how the final

capability prioritization decision might change if different values had been assigned. To accomplish this, an adaptation of an "extreme-case analysis," (also known as a "best-case, worst-case analysis") was chosen due to its relatively simple implementation in an AHP problem format [37].

For this analysis, maximum and minimum possible scores for each capability under the "Utility" and "Expected Difficulty" criteria were identified. Stepping through each capability, the entering assumption was that the nominal assigned score was incorrect. Recognizing this, an example follow on question for each capability becomes: "In a best-case scenario, how easy might this capability actually be to implement?" Similarly, the opposite side of this question is also pondered: "In a worst-case scenario, how difficult might the implementation of this capability actually be?" Similar questions were also posed for the Utility criterion, with best and worst case scores assigned for each potential capability. Also, note that the "# Scenarios" criterion was excluded from the sensitivity portion of this analysis. This was because the scores assigned to each capability for this criterion were more objective in nature since the score directly reflects the number of operational scenarios to which the capability contributes. The sensitivity values selected for each capability through this process, along with the original nominal scores, are shown in Table 3.7.

Table 3.7: Updated capability scoring matrix with max and min values (***Bold italics*** indicate dependence on other capabilities)

Capability	*Scenario Contribution*		*Utility Provided to User*				*Expected Implementation Difficulty*			
	# Scenarios Contributed	**Scenarios Score**	**Utility Provided**	**Nominal Utility Score**	**Min Utiity Score**	**Max Utility Score**	**Expected Difficulty**	**Nominal Difficulty Score**	**Min Difficulty Score**	**Max Difficulty Score**
Moved and setup by 1-2 technicians	1 2 3	3	Launch-Critical	3	2	3	Easy	3	1	3
Streamline setup and initialization	1 2 3	3	Optimizes Launch Process	2	1	2	Medium	2	1	3
Detect environmental parameters (wind data)	1 2 3	3	Optimizes Launch Process	2	1	2	Easy	3	1	3
Re-orient launcher if wind direction not favorable	1 2 3	3	Enhances System Interfaces	1	1	2	Hard	1	1	2
Disable launch capability if winds adverse	1 2 3	3	Enhances System Interfaces	1	1	2	Easy	3	2	3
Maximize launcher range envelope from GCS	1 2 3	3	Optimizes Launch Process	2	1	2	Medium	2	1	3
Detect people/objects in launcher vicinity	1 2 3	3	Optimizes Launch Process	2	1	2	Medium	2	1	3
Disable launch ability if launch area unsafe	1 2 3	3	Enhances System Interfaces	1	1	2	Medium	2	1	3
Automatic reset capability	1 2 3	3	Optimizes Launch Process	2	2	3	Medium	2	1	3
Lighting system to warn personnel of launch status	1 2 3	3	Launch-Critical	3	1	3	Easy	3	2	3
Safe to load indication	1 2 3	3	Enhances System Interfaces	1	1	3	Easy	3	2	3
Abort launch functionality	1 2 3	3	Launch-Critical	3	2	3	Easy	3	2	3
Mechanical-based kill switches with easy accessibility	1 2 3	3	Launch-Critical	3	2	3	Easy	3	1	3
Communicate launch system/sensor status to GCS	1 2 3	3	Enhances System Interfaces	1	1	2	Hard	1	1	2
Receive 'Halt Launch' command from safety observers	1 2 3	3	Enhances System Interfaces	1	1	2	Hard	1	1	2
Detect UAV on launch platform	2 3	2	Optimizes Launch Process	2	1	3	Medium	2	2	3
Disable launch ability until UAV loaded	2 3	2	Enhances System Interfaces	1	1	2	Medium	2	2	3
Identify UAV on launch platform	2 3	2	Enhances System Interfaces	1	1	2	Medium	2	1	3
Launch platform position sensors	2 3	2	Launch-Critical	3	1	3	Easy	3	2	3
Detect other launch system position/orientation	3	1	Enhances System Interfaces	1	1	2	Hard	1	1	2
Disable launch ability if oriented towards other launcher	3	1	Enhances System Interfaces	1	1	2	Medium	2	2	3
Alert technician of competing orentation problems	3	1	Enhances System Interfaces	1	1	2	Easy	3	2	3
Detect other launch system's launch status	3	1	Enhances System Interfaces	1	1	2	Hard	1	1	2
Disable if other launch system is in launch cycle	3	1	Enhances System Interfaces	1	1	2	Easy	3	2	3

The next step of this sensitivity analysis is to re-perform the complete AHP decision-making process using new combinations of the nominal, maximum, and minimum criteria values [36], [37]. For example, a complete analysis was performed using all nominal values except for the Utility Criterion, in which the Utility Pairwise Comparison Matrix was instead completed using the minimum utility values for each capability. Using these minimum values resulted in a new set of final AHP scores and a new capability prioritization ranking. Then, the process was repeated using the maximum values for the Utility Criterion producing another, different set of AHP scores and capability priorities. This entire process was repeated several times, first using nominal values for Scenarios and Utility with the minimum values established for the Expected Difficulty Criterion, and then again using the maximum difficulty values. Finally, AHP scores and capability rankings are tabulated using minimum values for both the Utility and Expected Difficulty criteria simultaneously, and then again using the maximum values associated with both criteria. A data table showing the scores for the capabilities under each set of assumptions is shown in Table 3.8.

Table 3.8: AHP sensitivity analysis scoring summary (***Bold italics*** indicate dependence on other capabilities)

Capability	Nominal Score	Utility Min	Utility Max	Difficulty Min	Difficulty Max	Both Min	Both Max	Average Score
Moved and setup by 1-2 technicians	0.069	0.064	0.060	0.064	0.066	0.059	0.057	0.063
Streamline setup and initialization	0.046	0.040	0.038	0.000	0.049	0.040	0.040	0.036
Detect environmental parameters (wind data)	0.051	0.045	0.043	0.046	0.049	0.040	0.040	0.045
Re-orient launcher if wind direction not favorable	0.032	0.038	0.036	0.034	0.033	0.040	0.037	0.036
Disable launch capability if winds adverse	0.039	0.045	0.043	0.039	0.037	0.045	0.040	0.041
Maximize launcher range envelope from GCS	0.046	0.040	0.038	0.046	0.049	0.040	0.040	0.043
Detect people/objects in launcher vicinity	0.046	0.040	0.038	0.046	0.049	0.040	0.040	0.043
Disable launch ability if launch area unsafe	0.035	0.040	0.038	0.034	0.037	0.040	0.040	0.038
Automatic reset capability	0.046	0.060	0.055	0.046	0.049	0.059	0.057	0.053
Lighting system to warn personnel of launch status	0.069	0.045	0.060	0.068	0.066	0.045	0.057	0.059
Safe to load indication	0.039	0.045	0.060	0.039	0.037	0.045	0.057	0.046
Abort launch functionality	0.069	0.064	0.060	0.068	0.066	0.064	0.057	0.064
Mechanical-based kill switches with easy accessibility	0.069	0.064	0.060	0.064	0.066	0.059	0.057	0.063
Communicate launch system/sensor status to GCS	0.032	0.038	0.036	0.034	0.033	0.040	0.037	0.036
Receive 'Halt Launch' command from safety observers	0.032	0.038	0.036	0.034	0.033	0.040	0.037	0.036
Detect UAV on launch platform	0.039	0.033	0.048	0.043	0.041	0.037	0.050	0.042
Disable launch ability until UAV loaded	0.027	0.033	0.031	0.031	0.029	0.037	0.033	0.032
Identify UAV on launch platform	0.027	0.033	0.031	0.027	0.029	0.033	0.033	0.030
Launch platform position sensors	0.061	0.038	0.052	0.061	0.059	0.037	0.050	0.051
Detect other launch system position/orientation	0.021	0.027	0.025	0.023	0.022	0.029	0.026	0.025
Disable launch ability if oriented towards other launcher	0.024	0.030	0.027	0.028	0.026	0.034	0.030	0.028
Alert technician of competing orentation problems	0.028	0.034	0.032	0.028	0.026	0.034	0.030	0.030
Detect other launch system's launch status	0.021	0.027	0.025	0.023	0.022	0.029	0.026	0.025
Disable if other launch system is in launch cycle	0.028	0.034	0.032	0.028	0.026	0.034	0.030	0.030

At the far right of this table, there is an "Average Score" column. Here, all the final AHP scores across each capability are averaged to yield a single value that reflects not only the original nominal score for that capability, but also the new scores generated as a result of the extreme case analysis of the subjective criteria measures. For ease of comparison, the original capability AHP scores and ranking are shown side by side with the new, sensitive AHP scores in Table 3.9.

Table 3.9: Nominal capability AHP scores with final sensitivity AHP scores (***Bold italics*** indicate dependence on other capabilities)

Capability	Nominal AHP Score	Sensitive AHP Score
Abort launch functionality	0.0689	0.0641
Mechanical-based kill switches with easy accessibility	0.0689	0.0628
Moved and setup by 1-2 technicians	0.0689	0.0628
Lighting system to warn personnel of launch status	0.0689	0.0587
Automatic reset capability	0.0464	0.0531
Launch platform position sensors	0.0615	0.0512
Safe to load indication	0.0393	0.0459
Detect environmental parameters (wind data)	0.0511	0.0448
Maximize launcher range envelope from GCS	0.0464	0.0428
Detect people/objects in launcher vicinity	0.0464	0.0428
Detect UAV on launch platform	0.0390	0.0416
Disable launch capability if winds adverse	0.0393	0.0411
Disable launch ability if launch area unsafe	0.0345	0.0377
Streamline setup and initialization	0.0464	0.0363
Communicate launch system/sensor status to GCS	0.0322	0.0357
Receive 'Halt Launch' command from safety observers	0.0322	0.0357
Re-orient launcher if wind direction not favorable	0.0322	0.0357
Disable launch ability until UAV loaded	0.0284	0.0317
Identify UAV on launch platform	0.0271	0.0303
Disable if other launch system is in launch cycle	0.0271	0.0303
Alert technician of competing orentation problems	0.0284	0.0303
Disable launch ability if oriented towards other launcher	0.0237	0.0283
Detect other launch system's launch status	0.0214	0.0248
Detect other launch system position/orientation	0.0214	0.0248

From this table, one can easily see that scores associated with some of the capabilities, such as "Moved and setup by one to two technicians," changed very little as result of this analysis. Others, however, such as "Automatic reset capability" or "Streamline setup and

initialization," saw fairly significant changes in their composite AHP scores as a result of the extreme case sensitivity analysis. All together, the above table provides a robust starting point for prioritizing and selecting potential launch system capabilities for development during an iterative prototyping process, recognizing that all three of the primary decision-making criteria are properly accounted for in the final sensitive AHP scores.

3.6 Capabilities Selected for Development

Based on the results of the previous analysis, the following capabilities were identified for development as part of the first launcher prototype:

1. Abort launch functionality
2. Mechanical-based kill switches with easy accessibility
3. Moved and setup by one to two technicians
4. Lighting system to warn personnel of launch status
5. Automatic reset capability
6. Launch platform position sensors

Capabilities expected to be explored during the development of the second prototype iteration include:

1. Safe to load indication
2. Detect environmental parameters (wind data)
3. Maximize launcher range envelope from ground control station
4. Detect people/objects in launcher vicinity
5. Detect UAV on launch platform
6. Disable launch capability if winds averse

Capabilities expected to be implemented during the development of the third launch system prototype include:

1. First and second-level capabilities not fully implemented in first or second prototypes
2. Disable launch ability if area unsafe
3. Streamline setup and initialization
4. Communicate launch system/sensor status to GCS

5. Receive “Halt Launch” commands from GCS or safety observers
6. Re-orient launcher if wind direction not favorable
7. Disable launch ability until UAV loaded

These capability groupings are created by identifying natural gaps in the Sensitive AHP Scores shown previously in Table 3.9. For example, when moving from “Launch platform position sensors” to “Safe to load indication” in this table, there is a notable drop in AHP score from 0.0512 to 0.0459. Due to the significant drop, this juncture was identified as a natural point to shift from first-iteration capabilities to second-iteration. Similar reasoning was used to identify natural separations between the second-iteration and third-iteration scores and capabilities, as well as those capabilities that were not expected to be pursued as part of this work. Additionally, potential capabilities that were originally identified but are not called out above for development as part a specific iteration were relegated to recommendations for future work. Such capabilities would definitely add value to a launch system, but were not pursued due to their lower prioritization values and time constraints for the completion of the total effort.

Finally, it should be emphasized that the capabilities and prioritization identified herein are only meant to be a starting point. As stated in [36]: “As with any structured decision-making process, the recommendations of AHP should not be followed blindly but carefully considered and evaluated by the decision maker(s).” In the context of a UAV launch system, this means that some mechanical system designs may necessitate the development of certain capabilities earlier than originally identified through this AHP decision-making analysis. With that understood, if a departure was made from the above capability sets for each iteration, the reasoning for those decisions were clearly highlighted and fully justified. Now, with all the potential capabilities identified, prioritized, and segregated, the design decisions and implementation of the first set of capabilities into the first launch system prototype can be reviewed.

CHAPTER 4: Rapid Launch System Prototype 1

4.1 Design Overview

The effort to design and build a rapid UAV launch system capable of supporting swarm operations for fixed-wing aircraft originated through group work conducted as part of Naval Postgraduate School (NPS)'s Systems Engineering capstone design courses. For these courses, a seven-person group worked for five months to conceptualize, design, build, and test a UAV launch system prototype to meet ARSENL's needs. Through this work, it was determined that in order for ARSENL to eventually execute an air war using swarming fixed-wing UAVs, they would need the capability to launch aircraft using a single launcher in a period no greater than 15 seconds [38]. The logic driving this target frequency stemmed primarily from ARSENL's stated goal of executing a 50 versus 50 UAV air war using swarming fixed-wing aircraft [38]. However, the Ritewing Zephyr II aircraft that ARSENL is currently using to work towards this goal only have a useful battery life of 45 to 60 minutes. As such, it was determined that in order for the ARSENL team to put forth an effective demonstration of swarming tactics in the air war, it would be necessary for all aircraft to be airborne in the first 15 minutes of the scenario to ensure sufficient battery power for all UAVs involved [38].

It quickly became obvious that, in addition to standardizing communications and implementing procedural changes regarding the preparation of UAVs for flight, enabling this 15-second launch rate would be a key design parameter for the new launch system [38]. This means that the new system either needs to be equipped for a fast but simple manual reset, or needs to be specifically designed with an automatic reset capability [38]. With this in mind, four potential launch system designs were conceptualized and, due in large part to its ability to be automatically reset with little to no operator intervention, a variation on a pneumatically powered launch system was selected [38]. An initial computer aided design (CAD) drawing of this system, officially named the Rapid UAV Launch Engine (RULE), is shown in Figure 4.1.

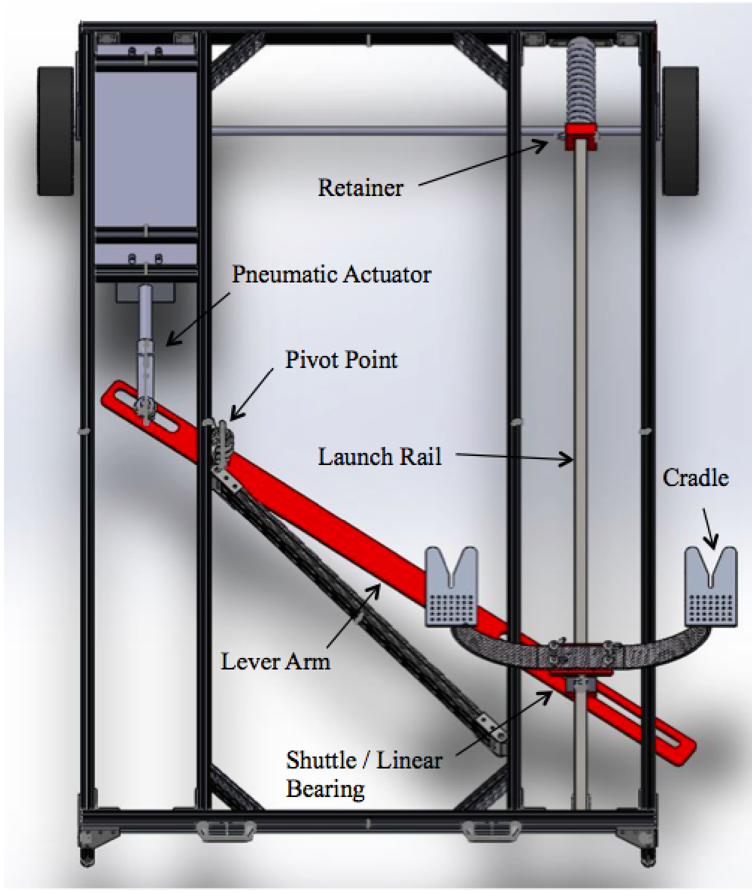

Figure 4.1: Overhead view of the Rapid UAV Launch Engine design, from [38]

Instead of using a complex pulley system to provide the mechanical advantage, as is done in the majority of existing pneumatically-powered UAV launch systems, the RULE instead incorporates a lever-arm and pivot system [38]. The launcher is designed for a pneumatic cylinder that, when pressurized, moves an actuator connected to the lever arm [38]. At the other end of the lever arm, a linear bearing is connected that carries the UAV interface and support platform as it travels down the launch rail [38]. The ratio of the lever-arm lengths on either side of the pivot point, in conjunction with the pressure of the air ported into the pneumatic cylinder, dictates the top speed the launch platform is able to achieve prior to reaching the end of the actuator stroke. The system is designed to be powered using an external air tank with an onboard compressor, and high pressure air is ported to the pneumatic cylinder using a three-way pneumatic valve. A final, but key, benefit of the pneumatically-powered lever-arm design is that the direction of air flow can be reversed in

order to facilitate a fast and easy reset of the system [38].

4.2 Design-Necessitated Requirements

As discussed in Chapter 3, the capability prioritization list developed through the AHP analysis is meant only to be a starting point. In order for any new launch system to be successful, key functions that enable the safe and repeatable operation of the primary system functions must be identified and developed early-on–even if those capabilities were not highly ranked in the prioritization process. With this in mind, recall that the capabilities originally identified for implementation into this first launch system prototype were:

1. Abort launch functionality
2. Mechanical-based kill switches with easy accessibility
3. Moved and setup by one to two technicians
4. Lighting system to warn personnel of launch status
5. Automatic reset capability
6. Launch platform position sensors

Note from this list that one of the most important capabilities identified through the AHP analysis, that is, the ability to perform an automatic reset, is also identified by the RULE design team as a key function [38]. With this in mind, one of the key benefits that the pneumatic lever-arm concept provides over competing design proposals is the potential ability to operate and reset the UAV platform using electrical signals rather than depending solely on human interactions with the system [38]. By utilizing a three-way pneumatic valve that is configured to be operated using electrically driven solenoids, it is possible both to operate and reset the RULE system remotely [38]. This potential for electrical actuation also made software based operation a much more feasible possibility, thereby facilitating easier implementation of several of the other enabling capabilities.

The mechanical design of the RULE launch system also necessitates the implementation of another of the above capabilities: launch platform position sensors [38]. When using pneumatic cylinders, it is generally preferred to avoid pressurizing the cylinder throughout its full length of travel. When this is done, the internal piston is caused to slam against the internal seals at one end of the cylinder or the other, causing the cylinder to wear much faster. Additionally, without some kind of sensor to notify the user or software system that

the UAV platform had reached the full launch or reset positions, the pneumatic cylinder would continue pushing on the lever for longer than necessary and, as a result, could cause distortion or damage to the lever-arm, pivot, or any number of other components [38].

Finally, due to the high speeds the launch platform is likely traveling as it accelerates the UAV down the rails for launch, it became clear to the RULE design team that some small degree of automation is required to ensure the pneumatic cylinder is sufficiently depressurized prior to reaching the end of the launch stroke [38]. Otherwise, it would be nearly impossible for someone to manually depressurize the pneumatic cylinder at the precise moment such that the aircraft would reach the required end speed without putting the system at risk of full pressurization through the end of the stroke. A computer and software system, however, are definitely able to detect the signals generated by the launch platform position sensors and de-energize the operating solenoids on the pneumatic valve almost instantaneously, thereby initiating a depressurization of the pneumatic cylinder and placing the system in a safe and stable state.

4.3 Hardware and Software System Design Priorities

Prior to selecting hardware components and software architectures for use in the implementation of these first capabilities, it was important to the RULE design team that the software and hardware systems already being used by the ARSENL team in their UAV swarm research efforts be understood [38]. The assumption here is if the RULE hardware and software architectures are developed to match the existing systems already being actively developed by the ARSENL team, such systems are more likely to be utilized due to a general sense of familiarity, and ARSENL team members are able to quickly and easily dissect and understand the system in the event that changes or capability enhancements are necessitated at a later date. Additionally, since the implementation of later, lower priority capabilities requires the ability to interface and communicate with ARSENL's existing computer and software systems, maintaining a degree of commonality between these systems and the RULE greatly simplifies the implementation and integration of said capabilities during later prototype development.

4.3.1 Software Systems

Currently, the ARSENL team develops and operates software primarily using the Linux Ubuntu operating system. Within this system, a robust library of directories and executable files exists that facilitates the operations performed by both the UAVs and the ground control station. ARSENL develops software primarily using the Python 2.7, Python 3.0, and C++ languages, and also has come to rely on the Robot Operating System (ROS) to perform certain key functions. ROS, at its essence, is an "open-source, meta-operating system" designed to facilitate the development and operation of robotic systems [39]. The ROS system itself is really just an environment in which a "distributed framework of processes (a.k.a. *nodes*) that enables executables to be individually designed and loosely coupled at runtime" [39]. The beauty of this system is that a wide range of hardware components, software drivers, and user-defined executable programs can all be easily integrated into one or more software suites in the ROS environment. This enables communications, data, and operating commands to be sent from nearly any connected device that can subsequently be received and interpreted by any other program, component, or device connected to that same computer system. As a result of this investigation, it was decided that the RULE's systems should also be designed to operate using the Linux Ubuntu operating system and the Robot Operating System, with software programming developed using the Python 3.0 language to the maximum extent possible [38].

4.3.2 Hardware Systems

The ARSENL team currently utilizes a wide range of commercial computer systems, autopilot, GPS, and hardware components in their swarm UAV development efforts. The component of primary interest to the RULE development team, however, was the embedded computer system–an ODROID U3 single board computer. "Originally, the team envisioned an embedded computer system fully incorporated into the physical design that would control software. However, in order to save time and arrive at a feasible solution, a preliminary solution uses a detached, standalone laptop with plans to integrate an embedded computer for a future iteration of the design" [38]. Thus, while it was recognized that a final launch system *should* be built to incorporate an embedded computer to simplify wiring and create a more elegant system, a personal laptop computer running the Linux Ubuntu operating system equipped with Python 2.7, Python 3.0, and the Robot Operating System ("Hydro"

distribution) was used to facilitate more efficient software system development in both lab and field environments [38].

Next, the focus was turned to lower-level hardware components that ultimately facilitate the implementation of capabilities selected for the RULE prototype. This includes the interfaces, sensors, and cabling that enables the computer and software systems to interact with and control the operation of the launch system mechanical components. One of the key recommendations from the ARSENL team-leader regarding the selection of these components was to give preference to component systems that have pre-developed drivers or application program interfaces (APIs) available for the Linux operating system to simplify software development efforts. The driving concept here was that it is better for the team to spend its time developing the software needed to get the components to simply work together in the desired manner rather than in decomposing and writing software drivers for each individual component [38]. Based on this recommendation, in addition to the RULE development team's experience through previous NPS coursework, the Phidgets line of sensors and control products were selected to form the primary interface between the computer and the RULE's mechanical systems [38]. The company manufactures and distributes a wide range of sensors, switches, actuators, relays, displays, and computer interface devices, and all components are well supported with an available Linux API and, in some cases, previous ROS integration packages. Finally, the components developed by the Phidgets company also lend themselves well to this application since most of their hardware systems are configured to connect to a computer using universal serial bus (USB) ports and cabling, a primary interface available on most modern laptop computers and on many embedded computer systems.

4.4 Capability Implementation

4.4.1 Automatic Reset

As previously discussed, one of the key benefits provided by the pneumatic lever-arm design is that it is operated and then reset by simply reversing the direction of high pressure air flow into the pneumatic cylinder. To facilitate this, a three-way pneumatic valve equipped with two electrically-actuated solenoids was selected [38]. These solenoids require 3.5 watts of power at 24 VDC to actuate, and need to be controlled using software to ensure

the system is depressurized before reaching the end of the launch stroke [38]. A number of Phidgets electrical relays are well suited for this application. For clarity, relays are essentially just electrically operated switches–an electrical signal causes the switch inside the relay to shift, thereby supplying or killing power to a connected component. A Phidgets Interface Kit was also chosen to provide the interface from the computer's USB ports to these relays.

Using this setup, launch system operation is initiated based on an input from the launch technician through a control device connected to the computer's USB port. For this control function, a Sony Playstation 3 Controller was connected to the laptop using a USB cable [38]. When the operator presses the correct button combination on the controller, the software sends a small electrical signal to one of the digital outputs on the Phidgets Interface Kit [38]. The launch system operating relay, which is connected to this digital output, is then shut, supplying 24 volts and 146 amps of DC power to one operating solenoid on the pneumatic valve. This causes the valve to shift from its normal de-pressurized position, and allows high pressure air from the external air compressor to flow into one end of the pneumatic cylinder while the other end of the cylinder is vented to atmosphere [38]. The actuator connected to this cylinder then moves outward, causing the UAV platform at the other end of the lever-arm to accelerate down the launch rail [38].

After the completion of the launch cycle and subsequent depressurization of the pneumatic cylinder and tubing, a similar sequence is initiated to reset the launcher [38]. In this sequence, a different button combination input causes a small electrical signal to be sent to another digital output on the interface kit [38]. This output is connected to a different relay, which then shuts and supplies the required 24 volts and 146 amps of DC power to the other solenoid on the pneumatic valve [38]. As before, this causes the valve to shift in the opposite direction, supplying high pressure air to the opposite end of the pneumatic cylinder and causing the UAV platform to move backward to its original, pre-launch position [38]. The system is then ready for a new UAV to be loaded and launched in the next operating cycle.

4.4.2 Abort Launch Functionality

The next capability added to the RULE was the ability to abort a launch after the launch process is initiated [38]. For the pneumatically-driven system, it made the most sense to implement this functionality by assigning a button to the operating controller that causes the switches in both the solenoid-operating relays to open [38]. Thus, in the event of an emergency, failed actuation, or an unusual system response, the operator can ensure that both solenoids are de-energized and the pneumatic cylinder is vented with a single press of a button [38].

4.4.3 Mechanical-based Kill Switches

For the RULE system, a small deviation was made for the implementation of the "Mechanical-based kill switches with easy accessibility" capability. It was determined that the spirit of this capability was to ensure that the operator would have a mechanical means of ensuring that the system would not operate–a safety mechanism, of sorts. Therefore, instead of implementing an electrical or mechanical "kill switch," the RULE team decided to create a subsystem that, when engaged, would prevent the UAV platform from moving - even if high pressure air is ported to the pneumatic cylinder [38].

To solve this problem, a retaining pin was added to the lever-arm that engages a mechanism that is, essentially, a car-door lock [38]. This locking system, which is shown in Figure 4.2, is capable of holding back more than 3000 pounds of pressure, and can be released by applying a small electrical signal to the locking device [38]. By wiring the lock mechanism to a third electrical relay, the user is provided with a means of applying or killing power to the lock, thereby allowing remote operation of the safety device using the same operating controller used to actuate or reset the launch system itself [38].

With this system in place, the RULE's reset function will push the launch platform back until the retaining pin on the lever arm engages the locking mechanism, at which point the system is considered "reset" [38]. From here, the operator will first need to load a UAV on the platform and then release the locking device before initiating the launch-stroke. Otherwise, a launch initiation command would still pressurize the pneumatic cylinder to full pressure, but would not cause any actual motion to take place unless the lock were subsequently released [38].

Figure 4.2: RULE mechanical holdback device, from [38]

4.4.4 Launch Platform Position Sensors

The next capability added to the RULE were the launch platform position sensors. As previously discussed, these sensors are a necessary addition so the computer and software systems will have a way of knowing whether the launch platform is in the "reset" or "launched" positions, thereby causing the system to open the electrical relays and depressurize the pneumatic cylinder. To implement this capability, the RULE team once again turned to components produced by the Phidgets company [38]. It was decided that the UAV launch platform would be embedded with a small rare-earth magnet, and two Phidgets magnetic sensors were added to one side of the launch-rail support structure [38]. One sensor was positioned towards the beginning of the launch stroke, to be activated when the launch platform reaches the "reset" position, and the other was placed near the end of the launch stroke, to be activated just prior to the launch platform reaching the spring-braking assembly [38]. This end-of-stroke sensor was also used to release power to the locking mechanism, causing it to return to the "engaged" position in preparation for the system reset operation [38]. Both sensors were wired to digital inputs on the Phidgets Interface Kit which, in turn, will cause a change in state for a software variable corresponding to the assigned digital input [38].

4.4.5 Moved and Setup by One to Two Technicians

While most of the capabilities implemented up to this point have been heavily reliant on software and computer systems, sometimes simplicity of execution is an advantage all of its own. To enable the RULE to be moved and set up by only one or two technicians, it is necessary to facilitate an easy means of system mobility. However, instead of adding unnecessary complexity to the system, and due in part to the relatively low weight (~100 lbs) of the system, a simple mechanical wheel and handle assembly is added to the RULE [38]. This system, shown clearly in Figure 4.3, enables the system to be maneuvered and positioned in much the same fashion as a wheelbarrow [38]. While a somewhat less elegant and much less automated solution than other capability implementations performed as part of this effort, the simple addition of wheels, an axle, and a pair of handles to the RULE facilitated a vital capability without requiring a significant investment of time or resources.

Figure 4.3: CAD drawing showing RULE's wheels and handles, from [38]

4.4.6 Capabilities Not Fully Implemented

One capability selected for first-prototype implementation through the AHP process that was not successfully completed in the RULE development efforts was the addition of a lighting system to warn personnel in the vicinity of the launch system's status. Ultimately, time and resource limitations dictated the abandonment of this effort, which started as the simple breadboard and light-emitting diode (LED) prototype shown in Figure 4.4 [38]. It was decided during the RULE development process that this capability will instead be refined and implemented during the construction of future launch system prototypes.

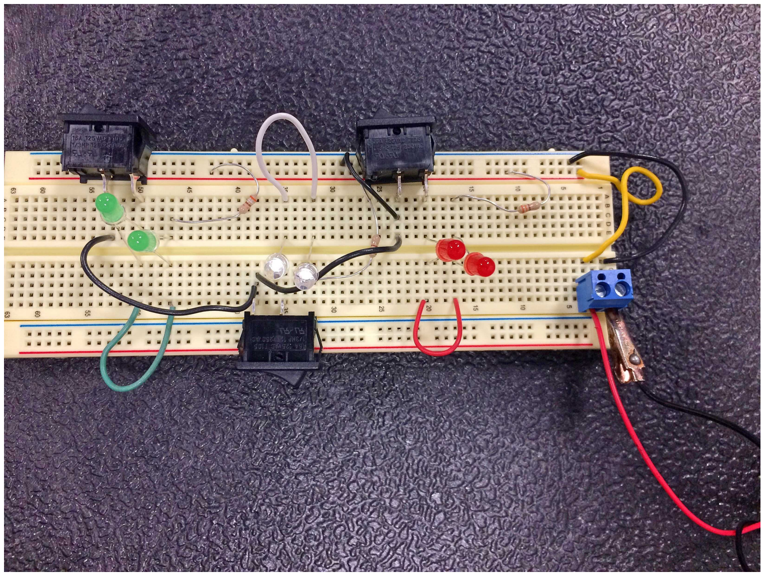

Figure 4.4: Intial prototype for the RULE LED communication system, from [38]

4.5 Electrical System Design

Having determined the means of implementation for each of the capabilities identified for the first launch-system prototype, it was necessary to create a wiring diagram showing how all the different components will be wired together to interact as envisioned. This diagram, shown in Figure 4.5 represents an important part of the system integration process–where all the electrical, mechanical, and software components are joined together to facilitate the desired system functionality.

Note that system was originally powered from a standard 115 volt AC power source, which is connected to a DC power supply that generates the required 12 and 24 volt DC sources needed to drive the onboard electrical components [38]. These 12 and 24 volt supplies each have a 25 watt load-resistor wired in parallel with the functional components to ensure a small load-current is drawn from the power supply at all times. The door lock mechanism is connected to the 12 volt supply, and is then attached to a Phidgets relay before finally tying into the power supply ground terminal. For operating the pneumatic cylinder, the 24 volt supply is connected to both the “Extend” and “Retract” electrical solenoids on the pneumatic three-way valve, and is then connected to two more Phidgets relays, one for each solenoid, before finally tying into the power supply ground terminal [38].

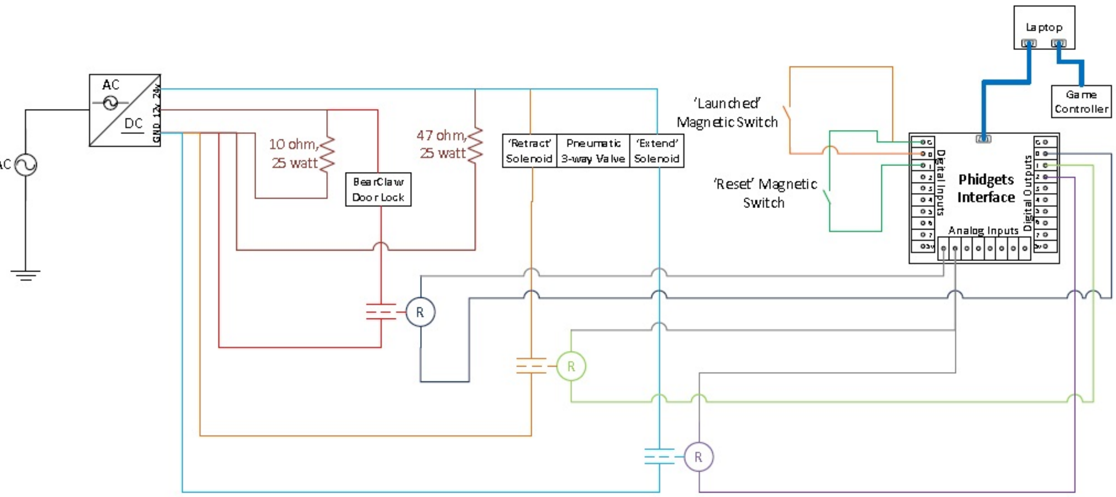

Figure 4.5: Electrical diagram for the RULE's electrical and computing subsystems

On the other side of the diagram, there is the laptop computer that is connected to both the system operating controller and the Phidgets Interface Kit via USB [38]. Digital outputs zero, one, and two on the Interface Kit are connected to the control ports on the three electrical relays, which are supplied power from analog input ports zero and one. Finally, the two magnetic switch sensors are connected to the Interface Kit using digital inputs zero and one. When the magnet embedded in the UAV platform passes by each of these sensors, the internal switch is shut and a logical "1" is supplied to the laptop's software systems from the Interface Kit variable corresponding to that input [38].

With this integrated electrical system design complete, the components were procured and the control circuits were built as outlined [38]. The final implementation of this integrated control system is shown in Figure 4.6 [38]. Here, the DC power supply is shown at the far right. At the top left of the image, the three electrical relays are shown which are then, in turn, tied to the electrical solenoids and lever-arm locking mechanism. The other side of the relays are connected to the Phidgets Interface Kit and finally, at the center of everything, there is an electrical breadboard used to tie all these inputs and outputs together.

4.6 Software System Design

At this point, the majority of the primary functionality of the RULE system and its sensors and computer-based capabilities have already been detailed. However, the architecture of

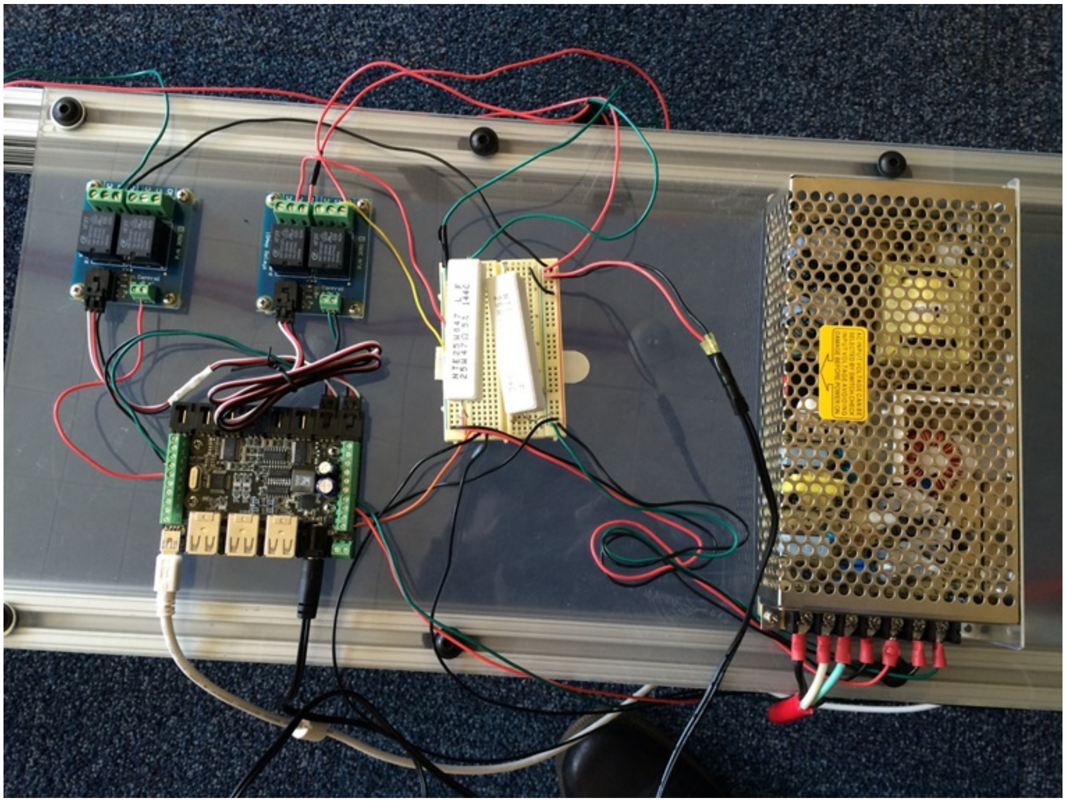

Figure 4.6: Overhead view of Rapid UAV Launch Engine's actual electrical subsystem

the underlying software systems that facilitate these operations should also be reviewed. As discussed previously, the software programming for the RULE was written primarily using the Python programming language, to be operated using the Robot Operating System on a Linux-based machine in order to maintain consistency with ARSENLs existing systems and architectures [38].

In ROS, executable files, called *nodes*, are connected and pass information using ROS's "message passing interface that provides inter-process communication" [39]. Each node in the ROS system either publishes or subscribes (or publishes *and* subscribes) to "messages" which contain pre-defined pieces of information [39]. Some of these nodes interact with software drivers to control or receive inputs from connected hardware components. Others simply process information from these messages and then, when a desired set of conditions are met, publish messages that call for certain responses driven by other nodes. As an example, the ROS communications diagram for the RULE system is shown in Figure 4.7.

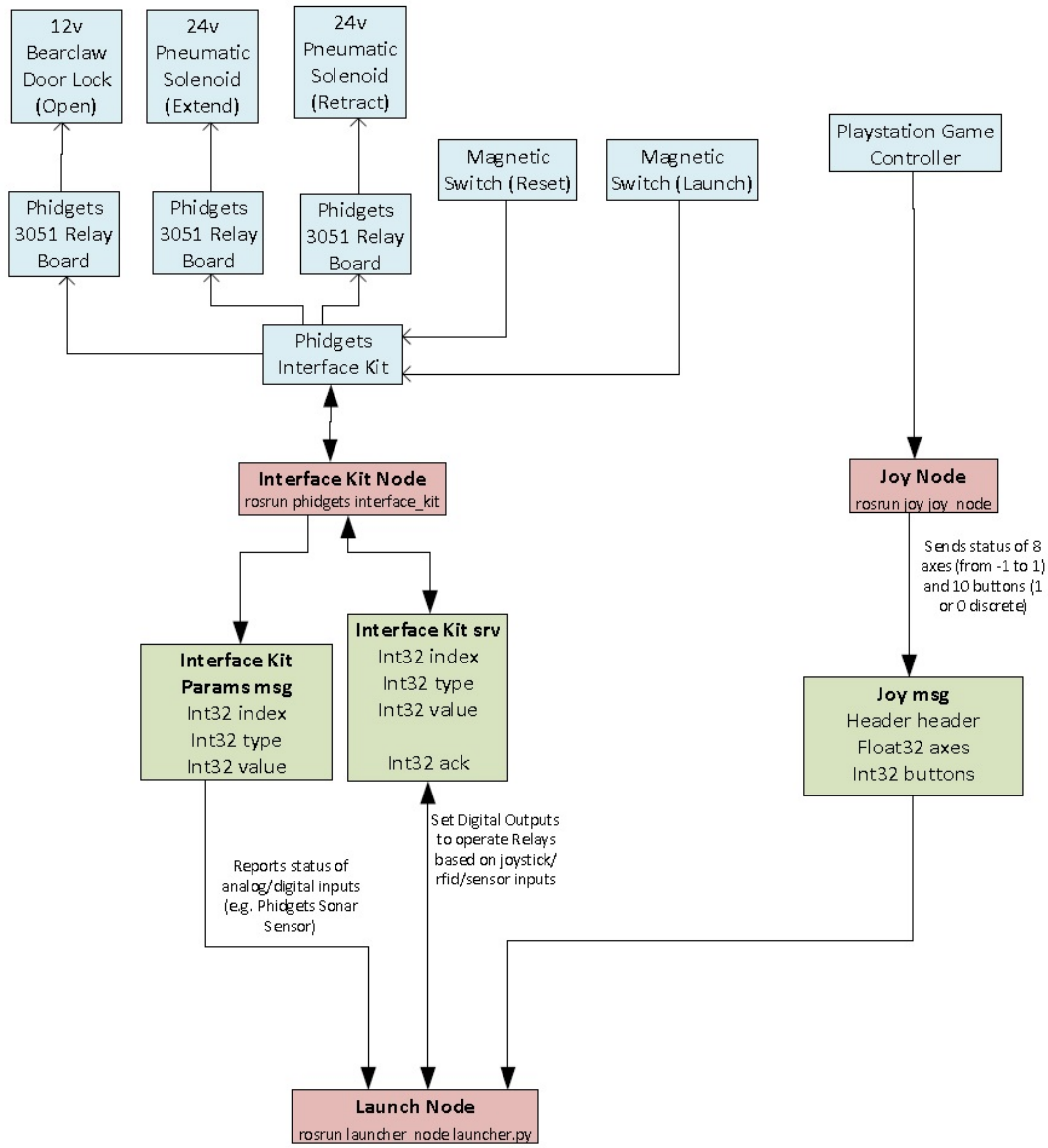

Figure 4.7: ROS communications diagram for the Rapid UAV Launch Engine

In this diagram, all hardware components are depicted at the top in light blue boxes. Starting with the Playstation controller, the USB interface interacts with the ROS system using an open-source set of drivers and executable files that detect button or trigger presses and then communicate those inputs to other connected components through the *Joy* node. This node then publishes the *Joy* message, which lists the state of every button on the controller,

through the ROS middle-ware. The *Joy* message is subscribed to by the *Launch* node script, which simply waits for a specific combination of buttons to be pushed before any action is initiated.

If the two triggers at the top of the game controller are pressed simultaneously with the "X" button, the *Launch* node places a ROS service call through the *Interface Kit* service topic. This service call sends a request to the *Interface Kit* node to actuate the digital output corresponding to one of the electrical relays. When this occurs, the relay shuts, closing the electrical circuit and supplying 24 volt DC power to the extension solenoid on the pneumatic valve.

Once the UAV platform approaches the end of its launch stroke, it will pass by the Phidgets magnetic sensing switch, thereby causing the switch to shut briefly and send a signal through the Interface Kit and into the ROS system via the *Interface Kit* node. In conjunction with operating the hardware drivers that enable the Interface Kit to function, this node also publishes information regarding the state of the digital inputs to the *Interface Kit params* ROS topic. The *Launch* node, which also subscribes to this topic and is awaiting this particular message, then places a second service call through the *Interface Kit* service topic. This second service call sends a request to the *Interface Kit* node to disable both the output corresponding to the launch relay and the output corresponding to the mechanical lock relay, thereby cutting power to the solenoid, de-pressurizing the system, and preparing the lock for retention functionality once the system is reset.

At this point, the system awaits a new command from the user via the game controller. If the button combination corresponding to the system reset function is detected, the software proceeds through a very similar set of processes culminating in the reset of the UAV launch platform. The functionality of the locking mechanism also follows a similar logic path. Finally, the software is configured so that any time, during any function the system could be placed in a safe state. This is facilitated by the "Triangle" button on the game controller. When pressed, the *Launch* node simultaneously sends service requests to disable all three electrical relays, thereby causing the system to fully de-pressurize and re-engaging the mechanical lock function.

4.7 System Testing and Conclusions

Once assembled, a number of lab and field-based operational tests were performed on the system, which is depicted in its final state in Figure 4.8. Through these tests, the operation of all electrical, software, and sensors-based functions were verified in a wide range of operating sequences and launch speeds. Ultimately, the RULE failed in its core objective of launching a Zephyr UAV, but that does not mean that valuable knowledge and insights were not gained through the design, building, and testing processes [38].

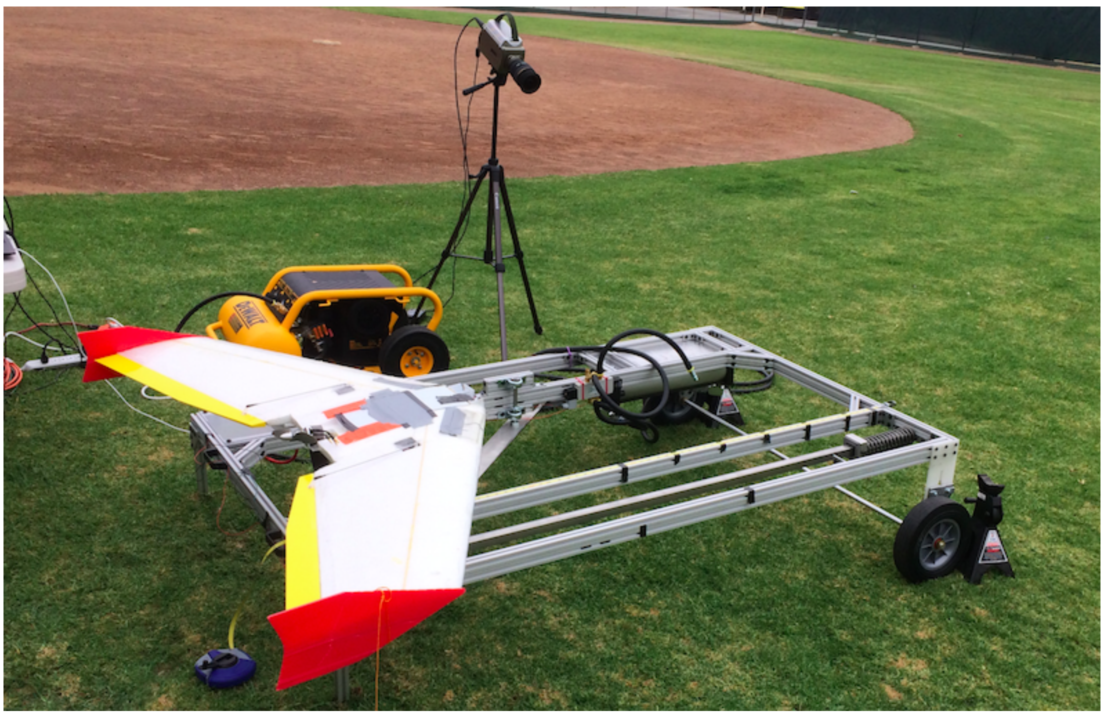

Figure 4.8: Final RULE system prepared for operational testing

On the positive side, the functionality of the software and electrical systems were generally successful. The system reliably responded to commands from the Playstation 3 game controller, and the launch platform moved forward and backward down the launch rail as desired [38]. The mechanical lock system was more than sufficient to hold back the lever arm, even with maximum pressure applied, and always released when commanded by the user [38]. Additionally, after some minor adjustments, the magnetic position sensors facilitated system depressurization during operation of the reset function in 100% of the trials conducted, and depressurized the system as desired during the launch stroke approximately

95% of the time [38]. Finally, the design team was able to demonstrate, on a small scale, the advantage that could be provided by a rapid launch system when engaging in swarm UAV flight operations through the RULE prototype [38].

The RULE system did have some downsides, however. First, and most importantly, it was unable to successfully launch any aircraft [38]. Second, despite the addition of the handles and wheels, the overall mobility of the system was somewhat limited when fully connected and pressurized due to the hose connection to the external air compressor [38]. This issue was exacerbated by the fact that all the onboard electronics were connected to a power supply that required an external AC power connection. With that said, it was generally no problem for a single individual to maneuver, set up, and initialize the entire system in less than 20 minutes. The system was also not reliable at high pressures and operating speeds [38]. When operating at maximum speed, the launch-depressurization function failed to activate a number of times, thereby leaving the system fully pressurized throughout the entire launch stroke and causing unnecessary wear or, in some cases, bending of structural components [38]. These problems were rarely seen, however, at lower operating speeds and air pressure settings. Finally, limitations on air flow rates and problems with the manner in which the UAV interfaced with the RULE's UAV platform were ultimately determined to be a significant contributors to the system's inability to successfully launch an aircraft [38].

The development of the RULE launch system had a significant impact on the design and execution of subsequent launch system prototypes. The primary mechanical objective of facilitating a UAV launch event and then having the system re-loaded and ready for launch in only a 10 to 15-second time span was reinforced and remained in place. However, it also became apparent that the proper design of other functions and interfaces is equally as important. The UAV attachment mechanism needed to be re-designed. The reduced maneuverability of the final system and tethers resulting from AC power requirements needed to be addressed. And finally, the reliability of key system functions at higher launch speeds needed to be addressed and enhanced. In the next rapid UAV launch system prototype, all these issues were tackled head-on. The experience and insights gained from the development of the RULE would serve the new two-man design team well.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:
Rapid Launch System Prototype 2

5.1 Design Overview

After completing work on the failed RULE launch system, ideation and brainstorming efforts quickly commenced to determine the form the next rapid UAV launch system might take. While several promising design options were formulated, the one chosen for development into the second launch system prototype consists of a DC motor coupled to a long length of roller chain using a set of rotational axles and toothed sprockets. For clarity, the roller chain and toothed sprockets that form the foundation of this system design are similar to those used to propel most modern-day bicycles.

The operating concept for the system is quite simple: a UAV is attached to the roller chain, the DC motor is powered-on, and the chain then accelerates the UAV to the opposite end of the launcher support rails, where it dis-engages from the roller chain and commences powered flight. The speed and power of the chain-drive system can be adjusted by changing the ratio of the primary and secondary sprocket sizes, or by changing the size of the DC motor or the voltage being fed to it. An initial CAD drawing of this new launch system prototype, unofficially dubbed the “Chain Launcher,” is shown in Figure 5.1. Additionally, a much more in-depth discussion of the processes and logic leading to the selection of this specific design are outlined by a sister effort entitled “Mechanical Design and Optimization of Swarm Capable UAV Launch Systems” [32].

As shown in the CAD conceptual drawing, this new prototype was constructed using cheaper and more readily available materials than the RULE launch system. The use of two-by-fours and PVC pipes to create the structure and support systems were intended to keep the system reasonably light-weight, yet still rigid enough to demonstrate the feasibility of the chain-launch concept. While simple in structure, this prototype ultimately underwent dozens of design and configuration changes en route to the final system configuration. This design evolution, as well as the supporting capabilities required at different stages in the process, are highlighted throughout the remainder of this chapter.

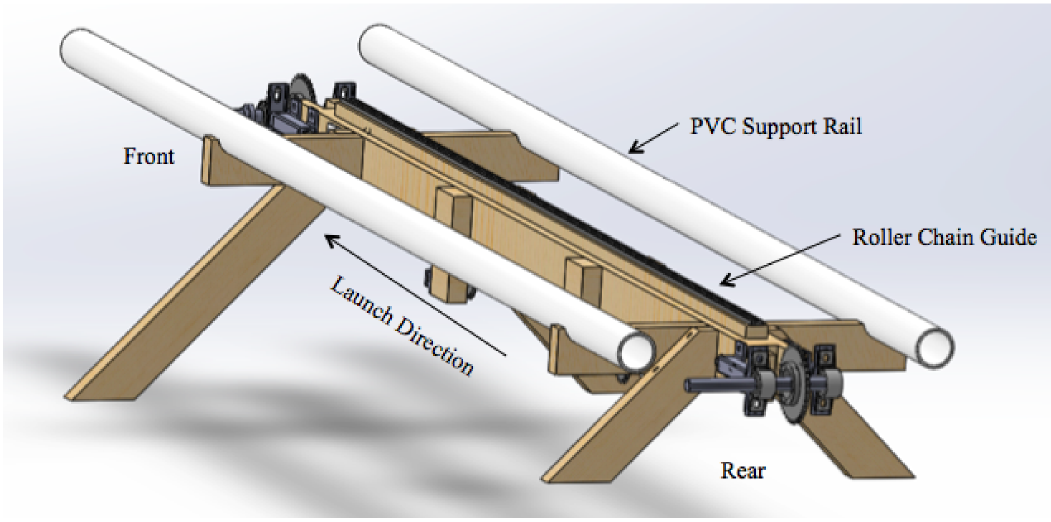

Figure 5.1: CAD view of the chain launcher design

5.2 Design-Necessitated Requirements

As with the RULE prototype, it is important to identify any key capabilities that are critical to the safe and repeatable operation of primary system functions. This process may alter the overall capability prioritization structure developed through the AHP analysis, but such changes are necessary if a system capable of successful field-demonstration is to be created. From Chapter 3, recall that the capabilities identified for implementation into this second launch system through the AHP analysis are:

1. Abort launch functionality *(Prototype 1 capability)*
2. Mechanical-based kill switches with easy accessibility *(Prototype 1 capability)*
3. Moved and setup by 1-2 technicians *(Prototype 1 capability)*
4. Lighting system to warn personnel of launch status *(Prototype 1 capability)*
5. Automatic reset capability *(Prototype 1 capability)*
6. Launch platform position sensors *(Prototype 1 capability)*
7. Safe to load indication
8. Detect environmental parameters (wind data)
9. Maximize launcher range envelope from ground control station
10. Detect people/objects in launcher vicinity
11. Detect UAV on launch platform

12. Disable launch capability if winds averse

Note that the above list includes the capabilities identified for implementation in the *first* launch system prototype as well as in the second. This is necessary since significant changes to the mechanical design for the launcher took place between the previous prototype and this one. Due to these design changes, the capability implementations completed during the development of the RULE system may not directly transfer to the new launch system configuration and, as such, are revisited during the development of the new prototype.

First, the Chain Launcher requires some means of actuating, stopping, and controlling the rate of acceleration for the DC drive motor. Theoretically, this can be done through some sort of direct software connection, or by using an electrical relay to apply or interrupt power to the motor in a similar fashion as that used in the RULE's pneumatic systems. Additionally, as previously identified, the ability to rapidly reset the UAV interface for subsequent launch events remains a key function for the Chain Launch system. However, unlike the RULE, an understanding of the exact position of the UAV launch platform is less important for a roller chain driven design since the UAV will simply detach as the chain and UAV spin around the sprocket at the end of the launch stroke. Since no time-critical functions are triggered from the UAVs's position in this particular design, such sensors were no longer required and were not included in the Chain Launcher prototype.

5.3 Hardware and Software System Design Priorities

Once again, it was important to the Chain Launcher design team that hardware and software architectures developed to operate the new launch system prototype be consistent with those already being employed by ARSENL in their own development efforts. As discussed later in both this chapter and in Chapter 6, this decision required significantly more effort on the part of the design team to learn and comprehend the existing software architectures and taxonomies. However, this effort paid dividends and proved to be highly beneficial when implementing later capabilities that required integration with pre-existing hardware and software systems. Due to this desire for consistency across platforms, the Chain Launcher software systems primarily operated on a Linux Ubuntu based computer system running the ROS middle-ware environment.

For hardware, the Chain Launcher design team acquired an ODROID XU single board computer and explored embedded computer implementation during the development of this prototype. While this is beneficial in the long run (as there is a learning curve when first starting to develop software using embedded computer systems), it was once again determined that a laptop computer ultimately provides a much better degree of flexibility during initial development and testing of new software systems. As such, a personal laptop computer running the Linux Ubuntu 14.04LTS operating system, which is equipped with Python 2.7, Python 3.0, and the ROS ("Indigo" distribution), is used for the development and operation of this prototype. However, software and capability implementations developed on the laptop computer are also continuously copied over to the ODROID computer's memory card to support an eventual shift to the exclusive use of an embedded computer system. Finally, the Chain Launcher design team chose to continue development efforts using the Phidgets line of sensors and interfaces. This decision was due largely to an established familiarity with the operation of these components and the fact that a robust set of drivers and a Linux API are readily available on the web.

5.4 Capability Implementation

5.4.1 Automatic Reset

The goal of facilitating an automatic reset for the new Chain Launcher prototype necessitated several configuration changes as the design of the system evolved. Initially, the prototype was designed to be operated in a manner very similar to the actuation of the pneumatic valve solenoids on RULE system. In this original configuration, a launch is initiated through an input received from the same Playstation 3 Controller used in the RULE. When the correct combination of buttons are pressed, the computer sends a signal to an electrical relay attached to a Phidgets Interface Kit. When this relay shuts, a 12 volt signal is provided to the operating coil of a DC contactor. A DC contactor is essentially just a relay, or electrically operated switch, used to pass or break higher voltages and currents than traditional relays or solenoids. For clarity, an example of the contactor used in the Chain Launcher and subsequent prototypes is shown in Figure 5.2. With the contactor operating coil energized, the contactor shuts, thereby providing either 12, 24, 36, or 48 DC volts to the chain drive motor. The motor then begins spinning, and the launch chain attached to the motor via a system of primary and secondary sprockets also begins to accelerate.

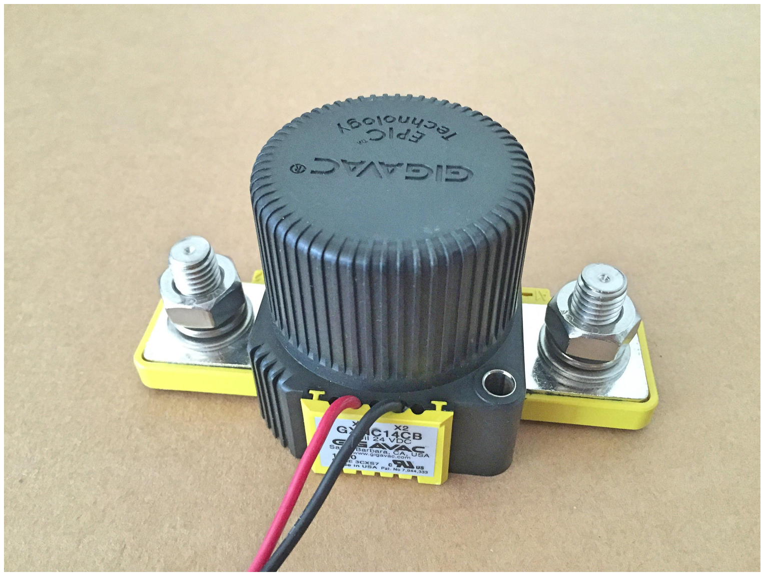

Figure 5.2: DC contactor used to provide software-based remote control of electrical power in the Chain Launcher prototype

The original plan for the Chain Launcher's reset function was, essentially, to design the chain-UAV interface such that the system is always ready to be loaded, regardless of the chain's exact position. To this end, the originally proposed launcher interface was a 3D printed plastic part that attached to the same hook on the bottom of the UAV that the original bungee-launch system used to accelerate the aircraft. Once affixed to the UAV, this part, which is shown in Figure 5.3, snaps into the small gaps between links in the roller chain and allows the aircraft to be accelerated during launch. At the end of the launch stroke, the toothed sprocket pushes the clip out of the roller chain and frees the aircraft for flight. With this interface, no actual system reset is necessary since, whenever the chain stops moving from the previous launch event, it is technically already "reset" and ready to accept attachment of another UAV. Unfortunately, however, this attachment system was unsuccessful since the printed plastic part lacked the holding strength required to keep the UAV attached as the chain began to accelerate [32].

At this point, it was determined that the UAV must instead be attached to the roller chain using a permanently mounted interface assembly, thereby re-generating the need for a true system reset capability. It was later determined that the time required to accelerate a UAV from rest to the desired launch speed of approximately 15 meters per second using a con-

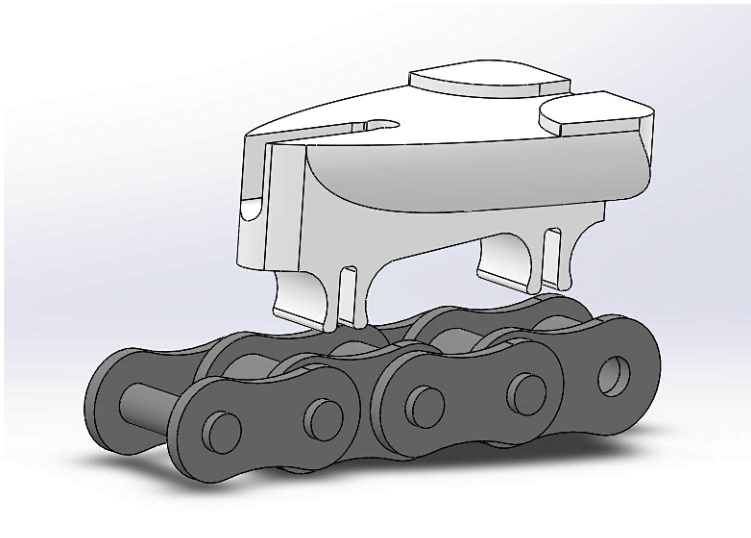

Figure 5.3: Original 3D printed UAV-chain interface clip

stant rate of acceleration would only require about half a second. With this vital piece of information, a new means of resetting the system was conceptualized - to provide power to the motor for a very specific and repeatable amount of time using the onboard computer and software systems. Technically, any time period longer than the 0.5 second launch time is sufficient to ensure the UAV is successfully launched, so the design team only needed to figure out the exact amount of time the chain requires to decelerate from full speed to a complete stop following launch. Then, by using the computer's system clock and some software programming to specify the exact amount of time that power was applied to the DC motor, an effective launch system reset was automatically executed as part of every launch cycle. While this tuning process timing took some trial and error to get right, it ultimately worked well enough that the UAV attachment interface was always reset to within about three inches on either side of of its ideal pre-launch position.

5.4.2 Abort Launch Functionality

The next capability added to the Chain Launcher prototype was the ability to abort a launch event after the launch process was initiated. Much like the RULE system before it, it made the most sense to implement this functionality by assigning a button to the Playstation controller that kills power to the DC motor by de-energizing the DC contactor. In a later modification to the Chain Launcher system, in which a Roboteq Motor Controller was

used to provide the operating voltage to the DC motor, the “kill” command was altered to simply command a motor speed of 0%. Thus, the Chain Launcher’s software and computer systems enabled an efficient means of quickly stopping the launch process with a simple, user-generated command.

5.4.3 Mechanical-based Kill Switches

As discussed previously, the Chain Launcher represented a major system redesign when compared to the original RULE prototype. Specifically, the new system was designed to be much more electrically-based in its fundamental operation and, as such, it made the most sense to implement an electrically-based, but manually operated kill switch. To accomplish this, a master power switch was installed in series with the DC motor and, eventually, to the corresponding Roboteq Motor Controller. This switch, shown in Figure 5.4, provides a physical means of ensuring that the system remains de-energized until the user is ready to operate it. The switch also provides the operator with a mechanical means of shutting down all system functionality at any moment. Finally, to ensure ease of access and un-obstructed operation in an emergency situation, the switch is both oversized and installed high up on the launcher support structure.

5.4.4 Moved and Setup by 1-2 Technicians

The RULE launcher used a simple mechanical solution to provide a moderate degree of system mobility. However, as discussed in Chapter 4, the mechanical-based mobility system, taken in conjunction with the RULE’s pneumatic hose and AC power cord tethers, resulted in some significant drawbacks when the RULE was set up and configured for operation. Future launcher prototypes, it was decided, should be freed from these constraints; thereby facilitating a truly mobile and highly adaptable system.

This new requirement for mobility during both the setup process as well as during system operation necessitated several prototype design developments. First, this updated mobility system required the integration of an independent electrical power supply system that was mounted onboard the launcher during operation. Since the main launch motor already requires 48 volts of DC power to operate, four 12-volt lead-acid batteries were affixed to the system support structure to provide power to both the launch motor and the mobility systems. This frees the system from any kind of electrical tether and makes it much more

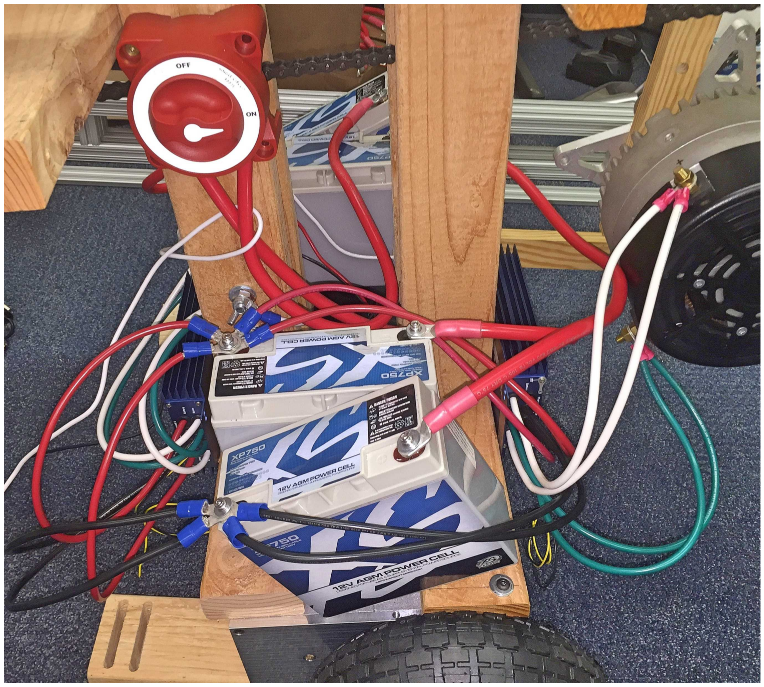

Figure 5.4: Electrical power kill switch for the chain launcher system

mobile and agile than the previous prototype. The unintended cost of this increased system mobility, however, is weight. At 10 feet long, and with a DC motor installed that weighed more than 35 pounds by itself, the Chain Launcher prototype is already significantly heavier and more difficult to manipulate than the RULE. The addition of the four, 15-pound lead-acid batteries to the assembly exacerbated this issue and, as a result, the new system's weight and configuration no longer lent itself well to a simple addition of wheels and axles.

Recognizing the potential mobility difficulties posed by the Chain Launcher's nearly 200-pound weight, as well as the myriad benefits that could be provided through a hands-off mobility solution, it was determined that the new prototype should be equipped with a motorized mobility system. To this end, two DC motors and gearboxes, which are shown in Figure 5.5, were selected and installed on the front-end of the Chain Launcher. These motors were then wired to the lead-acid battery array in much the same fashion as the main chain-drive motor, and a second Roboteq Motor Controller was added to enable fine-tuned control over the acceleration and speeds of the two wheel motors.

The design team quickly realized that, due to the launcher's new electrically-driven mobil-

Figure 5.5: Chain Launcher wheel, motor, and gearbox assemblies mounted to underside of launcher frame

ity functionality, continuing the use of a Playstation 3 controller tethered to the computer system via a USB cable was no longer a viable human-interfacing solution. Accordingly, efforts began to shift the controller from a wired to a wireless configuration. Initially, the plan was to simply re-configure the Playstation controller to communicate with the onboard computer via a Bluetooth USB dongle. However, after much trial and error, the design team discovered that establishing this Bluetooth communication channel is much more difficult than initially thought. To circumvent the problem, a new game controller that came packaged with a pre-configured Bluetooth USB dongle was purchased and installed. The new controller, shown in Figure 5.6, is a Logitech F-710 Wireless Gamepad that ultimately proved to be capable of true plug-and-play operation, thus facilitating wireless, un-tethered control of the Chain Launcher system [40].

While several other potential control interfaces for the launch system were investigated, the design team chose to stick with a game controller as the primary human interface device for a number of reasons. First, the Logitech F-701 gamepad provides a large number of buttons, triggers, and joystick devices. This enables the actuation or control of a wide range of functions and operations using the single control device. This large number of buttons also enables additional “deadman switches” for key functions to be implemented through software. For example, it is highly undesirable for the launch system to move during an aircraft launch event just because someone accidentally bumps one of the joysticks. To prevent this, a “deadman switch” is assigned in software that prevents actuation of the mobility subsystems unless a specific button is held down at the same time the joystick

Figure 5.6: Logitech F710 gamepad that controls the chain launcher system operation, from [40]

commands are given. Similar redundancies are also built into the software governing the actuation of both the launch and the slow-speed reset functions. Next, the potential to mount the controller in a custom-designed cradle during launch operations, while retaining the ability to remove that controller while maneuvering the system, made the wireless game controller interface an appealing option. Finally, the use of this control interface enabled the software-designer to assign certain control interfaces, such as the joysticks, to the mobility functions for which such an interface is best suited. Similarly, the actuation of stop-go type functions, such as the initiation of an launch event or the slow-speed reset functions, is easily accomplished through the use of simple button presses. A summary of all the operating commands for the Automated Multi-Plane Propulsion System (AMPPS) is shown in Table 5.1.

Table 5.1: Controller-based operating commands for the Chain Launcher and AMPPS prototypes

Function	Controller Operation
Launch UAVs	
Reverse Reset (Slow)	
Forward Reset (Slow)	
Operate Mobility System	

continued . . .

. . . Table 5.1 continued

Function	Controller Operation
Cancel Launch	
Remote System Shut-down	

5.4.5 Detect Environmental Parameters

During the development of the Chain Launcher prototype, the ability to detect environmental parameters, such as wind speed and direction, was also pursued by the design team. While the baseline capability is technically established during this iteration, the full functionality it provides was never actually integrated into the Chain Launcher's software systems. However, as the implementation of this capability represents the first major launch system integration effort with ARSENL's existing software and communications systems, the means by which this capability was facilitated should be highlighted.

Prior to beginning design work on this second prototype system, the ARSENL team had procured an Oregon Scientific WMR200a Professional Weather Center. This hobbyist-level home weather station comes equipped with temperature, humidity, wind chill, barometric pressure, wind speed, and wind direction sensors, and is shown in Figure 5.7 [41]. The weather station's sensors were configured to communicate wirelessly with a dedicated system console, which was then connected to a computer via USB to provide a real-time data

feed to the computer and, eventually, to outside systems as well [41].

Figure 5.7: Oregon Scientific WMR200A Weather Station used to collect wind data

Unfortunately, this weather station came equipped with software that only runs on the Windows operating system. Since the ARSENL team primarily uses the Linux Ubuntu OS on their computers, an alternative set of system drivers and data-logging software was needed to take advantage of the weather system's functionality. Internet searches regarding this problem led to the discovery of the open-source WeeWX weather station software package [42]. This free, Linux-based software suite contains drivers and support software for a number of home weather stations, and was originally developed to enable connection of these stations to computer servers for publishing real-time weather data to the web [42].

The next issue confronted was how best to integrate this weather system into the UAV rapid-launcher design. It quickly became obvious that mounting the weather station and console assembly onto the launcher itself would not represent the most elegant or useful solution to the problem. Instead, it was preferred to mount the weather station in a fixed, remote location in the vicinity of the ground control station. This remote location also ensured that reliable and accurate wind direction and speed data would be gathered at all times since the weather station and its console were mounted in a fixed position rather than on the mobile

launch system. A computer station was then set up nearby with the weather station console attached, feeding the data into the WeeWX software system via a USB connection. From here, the final obstacle was transmitting the weather data from the console and computer assembly to the launch system itself.

One of ARSENL's primary means of transmitting data and commands from the ground control station to the swarming UAVs is via a Wi-Fi network and custom data messaging system. Since both the launch system computer and the computer driving the weather station software were easily connected to this network through the addition of a Wi-Fi USB dongle, the addition of a new message type to the existing data messaging system enabled the periodic transmission of a weather data message over the network. This method of data transmission also provided an additional, unplanned benefit. Since the data message can be configured with any number of data points, temperature, humidity, and barometric pressure data can also be sent out over the network in addition to the wind speed and direction information. This facilitated a whole new way for the swarming UAVs to determine their precise altitudes - by using a real-time differential pressure calculation. Thus, the implementation of the environmental sensing capability for the launch system prototype facilitated new capabilities for existing systems already involved in the UAV swarming efforts.

5.4.6 Maximize Launcher Range from GCS

The ability to maximize the launch system's range from the ground control station was, essentially, facilitated through the design and implementation of the other capabilities highlighted in this chapter. First, the use of an onboard battery bank to drive the electrical systems and components eliminated the need for the launcher to remain close to an AC electrical outlet. Next, the Bluetooth-based wireless control system enabled the launcher to be remotely driven and operated at distances up to 30 feet away [40]. However, since the launcher is expected to be operated by a launch technician and not remotely by the GCS operator, this 30 foot range turned out not to be a significant limitation on the system since the launch technician should never be too far away. Finally, the implementation of the Wi-Fi-based communication system added to the overall mobility and range capability of the launcher since outdoor Wi-Fi networks are generally strong even at fairly moderate distances. Thus, the design and configuration of all the system capabilities identified up to this point actually work together to enable the launch system to be operated at fairly

significant ranges from the GCS.

5.4.7 Capabilities Not Fully Implemented

As the Chain Launcher prototype was only meant to be a proof of concept to prove the viability of the primary mechanical design, several capabilities identified for implementation into this prototype were either not pursued or, even if developed, were not fully integrated into this prototype. First, the launcher status lighting system was not developed for this prototype due to a lack of available space for the lights and general indecision over the physical form that this lighting system should take. In conjunction with this, no "Safe to Load" indications were implemented for this prototype due to the expectation that the warning light system might eventually assist in this functionality. Next, the launch platform position sensors originally utilized in the RULE system were abandoned for this prototype since the mechanical design for the Chain Launcher system reduced the necessity of exact position-sensing capabilities.

Several of the other first or second prototype capabilities were technically developed at this time, but were simply not integrated or installed onto the launcher itself due to time, space, or other constraints. The ability to detect people in the vicinity of the launcher was developed and bench tested during the creation of the Chain Launcher prototype, but the functions were never integrated into its design for operational testing and evaluation. The ability to detect aircraft loaded onto the UAV interface was also technically developed and bench-tested during the second-prototype build, but this capability was ultimately shelved for implementation on the third launch prototype instead. Finally, the ability to detect wind conditions and disable launch capability if these conditions are unfavorable is facilitated on the ground control side, but the software code that would enable the system to read and utilize this environmental data was never added to the Chain Launcher's computer systems.

5.5 Electrical System Design

Having identified the means of implementing each of the above capabilities, a new wiring diagram, shown in Figure 5.8, was created to map out the manner in which all the electrical components were wired together and powered for the Chain Launcher prototype.

As previously mentioned, all computing and software functions were executed using a stan-

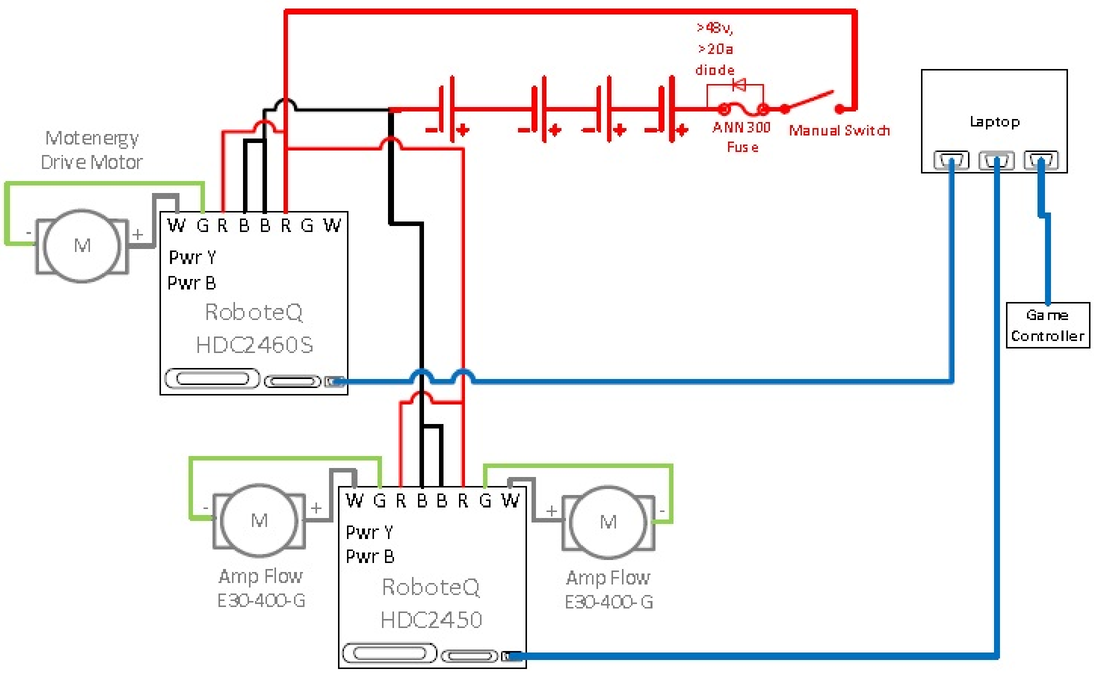

Figure 5.8: Electrical diagram showing operation of the Chain Launcher system

dard laptop computer. First, the Logitech F710 Gamepad's Bluetooth dongle is installed in one USB port to facilitate communications between the controller and the laptop. Two other USB ports were then connected to two different Roboteq motor controllers using standard USB cables and wiring.

The first motor controller, the Roboteq HDC2460S, was wired to the DC motor operating the roller chain and sprocket assembly. The second controller, a Roboteq HDC2450, has two separate controllable channels that were each wired to one of the motors driving the Chain Launcher's mobility system. Both motor controllers were then wired in parallel and connected to the lead-acid battery array. Starting with the motor controllers' black ground wires, the first battery array connection was to the negative terminal on one of the 12 volt lead acid batteries. The remaining batteries were wired in series, with the positive terminal of one battery connecting to the negative terminal of the next. The positive terminal of the fourth battery was connected to a Bussman-style 300 amp fast-acting fuse to protect the system from an unexpected overcurrent condition. The fuse then connects to one side of the main system power switch, and the other side of the switch was connected to the motor

controllers' red power wires.

With this electrical system setup, neither motor controller can operate until the main power switch is shut. Once shut, 48 volts of DC power is supplied to both motor controllers and powers them on. The motor controllers, after a five second software initialization process, then await speed commands from the computer system via a USB cable connection. When given a speed command, which is ordered in terms of percent power, the motor controller supplies the appropriate voltage to the corresponding motor. This continues until the controller receives a different speed command or until the main power switch is opened, thereby de-energizing both the Roboteq Motor Controllers and their associated DC motors.

These systems, operating at full power from an initially stopped condition, generate transient DC currents in excess of 200 amps. While running currents of only 50 amps or so are expected, it was nevertheless important to the prototype design team that all electrical components used in the wiring of these circuits be rated to amperages consistent with the peak starting current levels. As a result, all the wires shown in this diagram are sized to be eight American Wire Gauge (AWG) or thicker and are equipped with crimped ring terminals to ensure safe, reliable connections between adjacent components.

5.6 Software System Design

Having detailed the means by which the key system components associated with the Chain Launcher were integrated and connected together, attention can finally be given to the functionality of the underlying software systems. As a reminder, the majority of the software created in support of this effort is written using the Python programming language, and the ROS environment provides the primary means for facilitating a software-based interconnection of components. For clarity, the ROS communications diagram for the Chain Launcher system is shown in Figure 5.9.

In this diagram, the primary hardware components are shown in light blue boxes. Starting with the Logitech game controller at the top left, any button presses or combinations of inputs that occur on this device are communicated to the ROS system through the same open-source set of drivers and executable files originaly used to detect inputs from the wired Playstation controller. These drivers then communicate the status of each button on the controller, in real time, to other ROS connected components through the *Joy Node*.

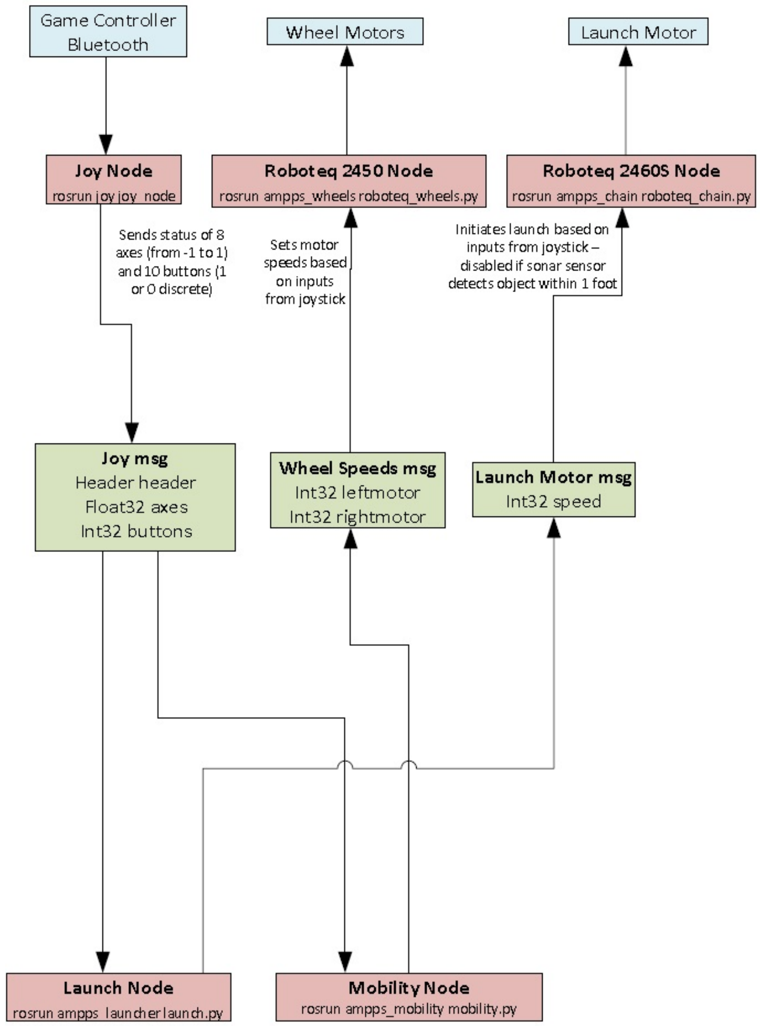

Figure 5.9: ROS communications diagram for the chain launcher system

This node publishes the controller's button status data to the *Joy* topic. The *Joy* topic is then subscribed to by both the *Launch Node* and the *Mobility Node*, which wait for specific button combinations to be pressed before any actions are initiated.

When a command to move the launch system is issued by holding the Left Trigger button while simultaneously moving the two controller joysticks (left = throttle, right = steering), the *Mobility Node* detects this condition and then publishes wheel motor power commands to the *Wheel Speeds* ROS topic. One nice feature of the Roboteq line of motor controllers is that they come pre-programmed to perform the signal mixing operations that are required to

create drive-able robotic systems. Thus, the throttle joystick's position is published as the "leftmotor" speed, the steering joystick's position is published as the "rightmotor" speed, and the *Roboteq 2450 Node*, which subscribes to the *Wheel Speeds* topic, communicates these motor speed commands to the Roboteq controller via a publicly available, open-source set of drivers and associated Linux API. This entire process repeats many times each second, transmitting the real-time positions of each joystick to the motor controller for conversion into individual wheel motor commands. Finally, it should be noted that the majority of the software used to create both this and the *Robteq 2460S Node* was created by another member of the open-source community, who originally adapted the Roboteq drivers and API for use as a ROS-compatible node in the publicly available "ros-roboteq-hdc2450" package. A few minor modifications to the scripts in this package ultimately enabled the design team to use this software to drive both the motor controllers at the same time while using the ROS interface.

Similarly, when a launch command is issued by holding the two buttons at the top of the game controller while simultaneously pushing the green "A" button, the *Launch Node* detects this condition and sends a single motor speed command to the *Launch Motor* ROS topic. This topic is subscribed to by the *Roboteq 2460S Node*, which then sends the commanded power, expressed as a percentage of the maximum available power, to the DC motor driving the roller chain assembly. To provide maximum control over the acceleration profile for the roller chain and attached UAV, a precisely-timed sequence of motor commands are issued when the "launch" command is given. These eight different commands, shown with the actual system response in Figure 5.10, are meant to step up the motor's torque in small increments over the length of the launcher, ensuring that the UAV never experiences acceleration forces in excess of three or four Gs. The result is a set of incremental, controlled increases in the actual speed of the roller chain and the attached UAV during launch. The system is then allowed to run for a pre-established amount of time until a motor speed of 0% is issued at the precise moment–thereby triggering the system to decelerate and stop with the UAV-roller chain interface located near the beginning of the launch rail.

Three other controller-initiated commands prompt a response from the roller-chain assembly through the ROS *Launch Node*. The first, a slow speed chain advance operation, is

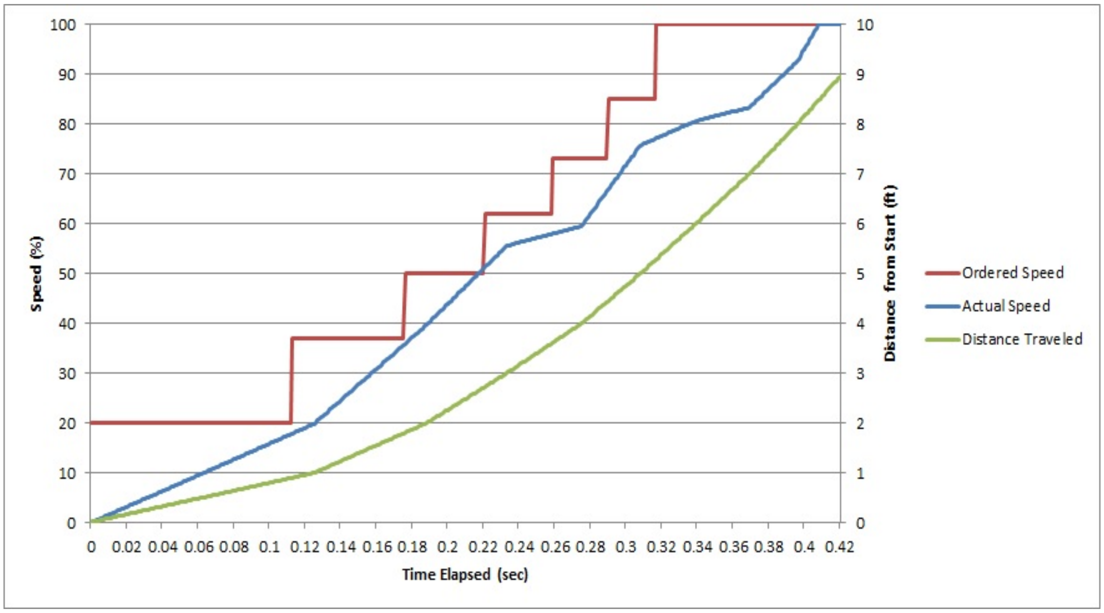

Figure 5.10: Chain Launcher roller-chain ordered and actual speeds during launch

actuated through another combination of two simultaneous button presses. This causes the chain to slowly advance forward until an interrupt command is issued. The second command operates in the same manner as the first, but instead causes the chain to slowly move in the reverse direction. These two functions enable easy fine tuning of the UAV-roller chain interface if the automated reset functionality is, for some reason, unsuccessful. The last command is a software-based system interrupt. If the yellow "Y" button is pressed at any time, motor speed commands of 0% are simultaneously sent to all three DC motors via both the *Launch Node* and the *Mobility Node*, thereby stopping all system motion. Similarly, if the computer system loses the connection to the Bluetooth controller, the system sends motor speed commands of 0% to all three motors until the connection is re-established.

The final portion of the software system that should be highlighted is the internal operating software for the two Roboteq Motor Controllers. These controllers are initially configured by the user through a proprietary software interface that facilitates nearly infinite control over the operation of any attached motors. Functions such as acceleration rate, deceleration rate, maximum operating voltage, maximum operating current, and dual-motor signal mixing (for creating drive-able robotic systems) are all available and easily configurable.

While not actually used for the Chain Launcher application, the Roboteq software also enables the user to create pre-defined operation scripts inside the motor controller itself, potentially alleviating the need for the ROS based timing functions which ramp up the speed of the roller chain according to the pre-determined acceleration profile. Ultimately, while expensive, the addition of the two Roboteq Motor Controllers was critical to enabling both the mobility and successful launch functionality for the Chain Launcher system.

5.7 System Testing and Conclusions

Unlike the RULE system, where system testing only took place at the end of the build process, the design of this second launch system prototype was tested and refined extensively from the very beginning until the final design shown in Figure 5.11 was reached. This process of test-redesign-test-redesign was critical to achieving and packaging all the desired functionality. Ultimately, the Chain Launcher was successful in its primary missions–it is able to be easily maneuvered, is capable of accelerating and releasing ARSENL's UAV at the desired launch speed, and is able to be configured for an automated reset through the use of software and precisely timed motor commands.

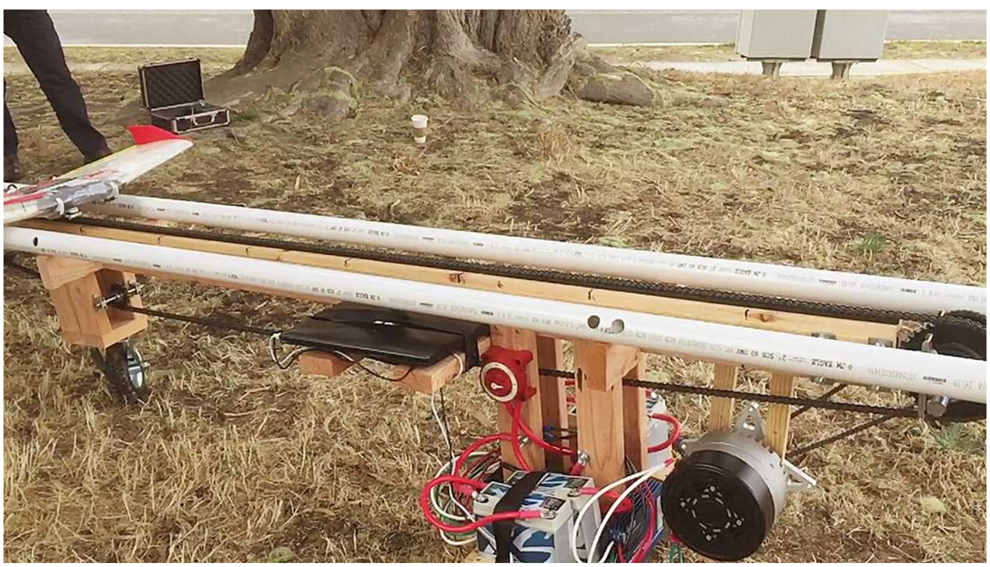

Figure 5.11: Final Chain Launcher system prepared for operational testing

While the Chain Launcher was eventually successful in executing all the base-level functions required of a swarm UAV launch system, the solution and packaging still need refinement. As discussed previously, many of the capabilities and functions identified for

implementation in this prototype were ultimately not incorporated due to time, space, or integration concerns. A fully developed launch system ultimately needs to incorporate many of these capabilities to facilitate the full range of functionality identified through the operational scenarios in Chapter 3. Additionally, the role that simple aesthetics can play in generating end-user excitement over a new system cannot be ignored. Thus, in addition to adding and integrating the remaining capabilities into the next launch system prototype, the implementation and wiring for the system also needs to be cleaned up and streamlined.

As with the RULE before it, the insights gained through the development of the Chain Launcher prototype and its supporting capabilities are critical to the successful design and implementation of the third prototype system. Since the second prototype is, essentially, a complete success with regards to the mobility, aircraft attachment, and successful UAV launch metrics, the new system is based largely on the final Chain Launcher design. This enables easy adaptation and integration of the key capabilities already achieved in the second prototype, but still allows for further design refinement as new capabilities and mechanical enhancements are built into the final, fully tested rapid-launch system.

CHAPTER 6: Rapid Launch System Prototype 3

6.1 Design Overview

Concept development for the third rapid-launch system prototype required significantly less ideation and raw brainstorming than the previous two prototypes. The successful demonstration of key launch system capabilities such as ability to attach and accelerate an aircraft, ability to release that aircraft for flight, the ability to automatically reset the aircraft-attachment interface, and the ability to easily maneuver and orient the launch system itself represented a huge first step towards the implementation of a fully functional launch system capable of supporting large-scale deployments of fixed-wing UAVs. With these successes in mind, the main priorities for the development of the third rapid-launch system prototype includes refining the overall construction of the Chain Launcher prototype, adding the remaining sensors, computers, and electrical-safety devices, implementing the remaining software-based capabilities, and optimizing the integration of all these systems and components into a functional, safe, and aesthetically pleasing launcher prototype.

The fundamental operation of this new system is very similar to the previous Chain Launcher prototype: a UAV is temporarily attached to an interface which is permanently affixed to a roller chain, a DC motor coupled to the same rotation axle as the primary chain sprockets is powered-on, thereby accelerating the UAV to the opposite end of the launcher support structure. Once the interface reaches the end of this structure, it commences traveling around the toothed sprocket and, in doing so, it dis-engages itself from the aircraft. The UAV is then left free to depart the launch system, using the momentum that has built up during the process to commence a gliding flight trajectory. After departure from the launcher, the UAV's computer and autopilot systems recognizes that minimum thresholds for speed and acceleration have both been met, and then actuate its onboard propulsion system. Meanwhile, the launcher's computer initiates the deceleration process, precisely timing the stop of the UAV interface to occur at its initial "reset" position. The initial concept design for this final prototype launch system, the AMPPS, is shown in a CAD drawing in Figure 6.1.

Figure 6.1: AMPPS launcher initial CAD design

The first, most striking difference between the AMPPS and the previous Chain Launcher system is the upgrade to an all-metal structure. The new system framing is constructed primarily out of extruded aluminum, and tension is adjusted using two brackets on each end of the launch assembly which moves the sprocket axles inward or outward. Over time, however, integration of components and implementation of new capabilities required further changes to the AMPPS initial design. Throughout the remainder of this chapter, many of these design changes are highlighted, with a focus on the necessity of such adjustments to facilitate effective system integration and increased overall capability for the final AMPPS launcher.

6.2 Design-Necessitated Requirements

In a now-familiar first step in the capability structuring process, the key capabilities that are critical to the reliable operation of the AMPPS launcher must initially be identified. From Chapter 3, recall that the capabilities identified for implementation into the the third launch system prototype from the AHP analysis were:

1. First and second-level capabilities not fully implemented in first or second prototypes
2. Disable launch ability if area unsafe

3. Streamline setup and initialization
4. Communicate launch system/sensor status to GCS
5. Receive "Halt Launch" commands from GCS or safety observers
6. Re-orient launcher if wind direction not favorable
7. Disable launch ability until UAV loaded

In addition to those capabilities selected for exclusive implementation into this third launcher prototype, a key advantage of the iterative prototyping process is that capabilities developed during previous iterations can be adapted and included into subsequent designs. On the flip side, for prototype iterations with similar designs, an unintended side effect of this prototyping process can be the propagation of general design weaknesses. As such, since the AMPPS launcher is essentially just an adaptation of the previous Chain Launcher design, many of the weaknesses and potential issues inherent in that second prototype also likely apply here.

First, as with the previous two launch system prototypes, the AMPPS requires a means of starting, stopping, and controlling the position and acceleration rate of the roller chain attached to the DC motor. The AMPPS also requires a means of resetting the launcher's UAV interface at the end of each launch event. Furthermore, to be a truly mobile and adaptable system, the AMPPS also needs an adequate, easily transportable electrical power supply. Finally, as it is primarily constructed out of extruded aluminum, the AMPPS system may likely suffer from the same (and probably more pronounced) weight excesses that plagued the Chain Launcher. Fortunately, solutions for all these issues have been successfully implemented and field-tested in the previous launch-system prototype and, as such, need only minor alterations to be properly integrated into the AMPPS system.

6.3 Hardware and Software System Design Priorities

As with previous iterations, the AMPPS design team desires to develop hardware and software architectures that are consistent with those already in use by the ARSENL team. At this point in the prototype development process, however, this decision was less significant in its implications than it was for the previous two designs. Since this same priority has already been established during the development of the RULE and the Chain Launcher prototypes, the design team already developed a moderate degree of comfort and familiarity

with the operation of the Robot Operating System, the Linux Ubuntu operating system, Phidgets sensors and interface components, ODROID single board computer systems, and the Python programming language. As such, these same computer, software, and hardware systems are once again selected for use in the AMPPS launcher, enabling the design team to continue taking advantage of this established and growing base of knowledge.

As with the Chain Launcher, it is decided that, to enable easier developmental testing of software and computer systems, a parallel development strategy was optimal. This means that software was simultaneously developed and implemented on both a personal laptop computer *and* on an ODROID XU single board computer system. Parallel development was intended to facilitate increased flexibility during initial development and field testing, but also enabled a quick and easy transfer to an onboard embedded computer once the software-side implementation of launcher capabilities had sufficiently matured.

6.4 Capability Implementation

6.4.1 Capability Implementations from Prototype 2

Due to the similarity of the AMPPS to the previous Chain Launcher design, many of the capabilities implemented through the second prototyping effort can be directly transferred to the AMPPS with little-to-no modification. For instance, as previously identified, the ability to attach an aircraft to the roller chain, accelerate that chain, release the aircraft, and then return the UAV interface to the original "reset" position had already been successfully demonstrated. Thus, to implement this same capability on the AMPPS, the only requirement was to transfer the DC motor, lead-acid batteries, electrical cabling, electrical switches, and the Roboteq HDC2460S Motor Controller over from the previous prototype. Thus, as with the Chain Launcher, the automatic reset function was facilitated through software timing and the use of the Roboteq motor controller to enable fine-tuned operation of the primary DC drive-motor.

As with the automatic reset function, the ability for the operator to issue a command to abort the launch process was already built into the software used in the Chain Launcher system and, therefore, the transfer of this functionality into the AMPPS was a relatively painless process. Once again, the operating concept was that the user presses a specific button on the Logitech controller which, at any time, tells the software to stop all motors.

This "kill" command was then converted to a motor speed command of 0% which was transmitted to both Roboteq motor controllers, thereby stopping any motion. While this worked well initially, the abort-launch command became somewhat less effective following the integration of the launcher status lighting system. At this stage, when a launch sequence was initiated, a short, audible countdown occurred prior to launch. Unfortunately, the abort launch command was currently disabled while this countdown was in progress. However, since this emergent issue was caused by a weakness in the software logic, the AMPPS design team intends to address and correct this problem during future system development efforts.

In addition to the existing, software-based abort-launch capability, an additional safeguard against undesired system operation was implemented during the construction of the AMPPS prototype. A normally-shut DC contactor was wired in series with the battery array and the two Roboteq motor controllers. As a reminder, a contactor is essentially an electrically operated, high power switch that operates in a similar fashion as an electrical relay. The operating coil for this contactor was then connected to a Phidgets relay which was driven by a digital output on the Phidgets Interface Kit. When a full system emergency stop is called for by the operator via the Logitech controller, a small electrical signal is provided to the Phidgets operating relay, thereby energizing the contactor coil and causing it to interrupt the current flow from the battery bank to the Roboteq motor controllers. This causes both controllers to shut down, placing the entire system in a safe, de-energized state until the contactor is reset by the operator.

The next capability brought in directly from the second launcher prototype was the ability for the system to be moved and set up by no more than one to two technicians. Recall that, due to the substantial weight of the Chain Launcher system, motorized wheels were added to facilitate remote-controlled movement and positioning. Since the conversion of the wooden support structure to an all-metal frame in the AMPPS was unlikely to yield any significant reduction in the total system weight, the motorized wheels and associated control systems again become a key design priority to ensure ease of mobility. Fortunately, the familiar design of the AMPPS system again made this transfer of capability a relatively painless process. Thus, the ability of the AMPPS to be moved and set up by no more than one to two launch technicians was ultimately facilitated through the use of mo-

torized wheels, the wireless Logitech control device, computer software and programming systems, and a Roboteq HDC2450 dual-channel motor controller.

The ability of the Chain Launch system to detect environmental conditions in real-time represented a significant improvement in UAV launch system capability. Unfortunately, while the ability to detect, transmit, and receive this weather data over the Wi-Fi network was technically facilitated and independently tested in the previous iteration, the ability to internalize and act on this data was never fully integrated into the final Chain Launcher design. As such, the original plan for the AMPPS system was to integrate and fully implement this capability. With the weather station already set up and piping data out over the network, this integration effort first required the development of the ROS nodes and executable files on the launcher system that would communicate the weather station data to other ROS nodes for processing and action. This functionality is currently implemented in the software onboard the AMPPS. However, while the launch system has the ability to receive the Wi-Fi weather-data message, the team fell short in developing the ability to convert these real-time Wi-Fi messages to ROS messages for use in that environment. As a result, the AMPPS launcher, as currently tested, is unable to adapt to real-time changes in weather conditions. For now, an arbitrary set of wind parameters are hard-coded into one of the ROS nodes, which then publishes this data for simulated interpretation by other nodes and programs.

This discussion of the Wi-Fi network over which the AMPPS is expected to communicate provides an excellent starting point for discussing the implementation of the final Prototype 2 capability: maximizing the launcher's operating envelope from the GCS. As with the Chain Launcher, this maximum operating range was first optimized through the use of an onboard battery bank and a wireless Bluetooth control device. These key design decisions facilitated a system where the only significant range-limitation was the ability to communicate with the GCS over the Wi-Fi network. Furthermore, if the capabilities provided through these Wi-Fi communications were no longer required, the system could theoretically travel as far as its batteries will carry it, so long as the operator remains within the 30 foot range of the Bluetooth controller. Thus, the AMPPS system is technically able to operate at limitless ranges from the GCS, albeit with reduced capability at significantly longer distances.

6.4.2 Mechanical-based Kill Switches

As discussed in Chapter 5, the Chain Launcher prototype utilized a single, manually-operated electrical switch to provide a software-independent system shutdown capability. This switch also served as a general ON-OFF switch for the onboard electrical components, and was oversized to ensure ease of identification and operation in the event of an emergency. While generally sufficient for the less refined Chain Launcher prototype, the AMPPS design team desires a more elegant, easily accessible, and individualized approach to this problem for the next prototype iteration. To this end, the switch panel shown in Figure 6.2 is created.

Figure 6.2: AMPPS mechanical-based system kill switches with easy operator accessibility

In this figure, an electrical system control panel with switches corresponding to the three primary subsystems was mounted at the rear-end of the AMPPS launcher. The three switches correspond to the computer and sensor subsystem, the wheels (or mobility) subsystem, and the primary launch motor subsystem. The system was also equipped with the same master power switch as was used in the Chain Launcher. The idea here was that, even if primary system power is being supplied from the batteries to the electrical components

through the master power-switch at the front end of the launcher, both the launch motor and mobility subsystems would be unable to operate unless their corresponding power switches are in the ON (or UP) position. These switches, like the master switch, are software-independent and require physical interaction from the launch technician to operate, but also provide an easily accessible, physical means of stopping each independent subsystem should an emergency situation arise. Furthermore, they add a degree of redundancy to the design since two independent, physical switches are required to be manipulated to enable activation of the launch motor and mobility subsystems.

A final benefit to using these small switches at the rear of the launcher rather than routing the single master power switch was the weight and space savings. The eight AWG wire that powers the primary, high amperage components at the front end of the launch system is significantly thicker, heavier, and more expensive than the 18 AWG wire required for the three-switch system. This added cost and weight provides little, if any, benefit to the user in terms of increased capability or performance and, as such, the independent switch system further distinguishes itself as the most effective solution to this problem.

6.4.3 Detect Personnel in Launch Area

The next noteworthy capability enhancement developed and tested in the AMPPS launch system was the ability to detect personnel and objects in the vicinity of the launch area. The goal of this capability would be to facilitate disabling the “Launch” function should an object or person be detected anywhere at the front side of the launch system, or within two feet of the rear side of the system. This provides an operator-independent means of ensuring personnel safety prior to a launch initiation. To facilitate this capability, two Phidgets MaxBotix sonar sensors with range capabilities out to approximately 25 feet were obtained. One of these sensors, located at the front end of the AMPPS launcher, is depicted in Figure 6.3.

These sonar sensors each connect to an analog input port on the Phidgets Interface Kit, and are also connected to digital output ports on the Interface Kit to allow for future iterations of the launch system software to turn these sensors on and off, if desired. The raw data measurement fed in through the sensor is published to the ROS *Interface Kit* topic, which can then be pulled into other ROS nodes for conversion and manipulation. When a sensor is dis-

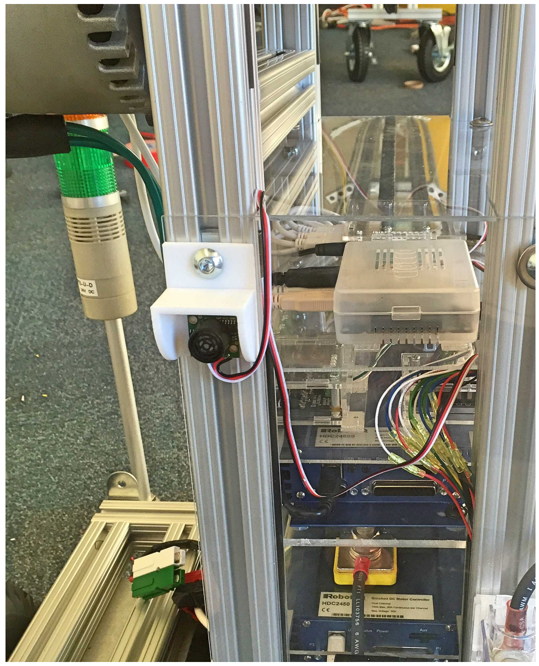

Figure 6.3: A personnel-detecting sonar sensor at the front of the AMPPS launcher

abled by the Interface Kit's corresponding digital output, the software is pre-programmed to return a range value corresponding to the maximum 25 foot operating distance. This essentially tells other ROS nodes requesting the sensor data that no personnel or objects are detected.

6.4.4 Disable Launch Ability if Area Unsafe

Having developed the ability to detect personnel and objects at both the front and rear of the AMPPS launcher, the software logic required to automatically disable a launch actuation can finally be implemented. To accomplish this, minor changes were made to the *Launch* node in ROS which requires both sonar sensor readings to be greater than pre-defined distance values in order for the "Launch" command to be transmitted to the drive motor. If the front sensor detects any objects within its 25-foot useful range, or if the rear sonar sensor detects any objects within two feet of the rear side of the launcher (indicating the operator is standing too close), the software system will prevent a launch command from being transmitted to the primary launch motor.

While this capability was fully implemented and tested through the AMPPS development

effort, it was ultimately disabled in the interest of ensuring consistent, reliable operation. While the sonar sensors do successfully identify personnel and provide accurate range values to the software system, they are also highly prone to false-positives which can impede effective employment of the system. As a result, instead of allowing software to automatically disable launch functionality, for the majority of the AMPPS's operational testing the sonar sensors were simply used to communicate an undesirable condition to the operator using the new launcher status lighting system. Thus, while technically implemented in full through this effort, the "Disable-launch" capability requires further refinement through new software algorithms or hardware choices to ensure the functionality provided is both reliable and accurate.

6.4.5 Launcher Status Lighting System

The addition of a launcher status lighting system was another simple, yet highly useful enhancement. Such a system enables the AMPPS to communicate the status of key system parameters directly to the launch technician in real-time without requiring a direct interface to the computer. Instead, for example, when a person is being detected within the pre-defined *Danger* range of the sonar sensors, a solid red light can be powered-on to alert the launch technician that an anomaly is being detected by the computing and sensing systems. Recognizing the potential usefulness of this capability, the attention turned to the means through which this capability would be implemented. Small, LED lighting systems at the rear of the launcher were considered, as were larger, flashing lights like those that are seen on unmarked police or emergency vehicles. A variety of industrial lighting systems were also investigated but, ultimately, the 24-volt light tower shown in Figure 6.4 was selected for integration into the AMPPS system.

This light tower was selected for several key reasons. First, the lights are big and bright enough to be seen not only by the technician at the rear of the launcher, but also by any personnel working in the system vicinity. Additionally, at only 20 inches tall, the light tower is adequately sized for appropriate, unobstructive installation onto the AMPPS base without hindering the installation or operation of any other components or systems. The tower is also wired to operate on 24 volts of DC power, enabling easy integration with the AMPPS's existing electrical power systems. Finally, each of the lights on the tower are wired for independent operation, allowing for myriad communication sequences to be

Figure 6.4: AMPPS status-indicating lighting system

implemented through software with no additional wiring or hardware requirements.

The integration and control of this lighting system requires the use of the Phidgets Interface Kit and four Phidgets solid-state relays. The tower is powered from a 24-volt connection to the battery array. The individual components on the tower, that is, the audible alarm and the red, yellow, and green lights, are each connected to a Phidgets relay which, in turn, links each component to the ground bus. When a light or audible tone is called for by one of the ROS software nodes, the corresponding digital output on the Interface Kit is triggered, actuating the relay and providing a closed path to ground through the appropriate component. The light or tone then turns on until this signal is interrupted and the relay re-opens.

For the purposes of initial AMPPS operational testing, the following light sequences are defined:

- Steady Red Light–Sonar sensors detect person or object in a *Danger* area
- Flashing Red Light–Wind speeds are greater than two knots and the launcher's orien-

tation, as detected by a Phidgets magnetic compass and spatial sensor, is not within 90 degrees of the current wind direction

- Steady Yellow Light–A UAV is detected on the launch platform interface by the AMPPS's radio frequency identification (RFID) reader
- Green Light–A launch has been initiated by the technician and is either in the countdown stage, in progress, or has just occurred and the roller chain is currently spinning down
- Intermittent Beeping Tone *(three beeps)*–A launch event has been initiated by the technician and a three second countdown is in progress

Ultimately, these sequences are flexible and easily available to change. They are established merely to provide a demonstration of capability during initial operational testing and, as a result, will likely change as the ARSENL team determines the parameters and sequences that are most beneficial to their processes in the field.

6.4.6 Safe to Load UAV Indication

The safe-to-load indication is technically facilitated through the addition of the launcher status lighting system, but was not fully implemented as part of this effort due to ambiguity regarding how this particular parameter should be defined. Technically, several redundancies were already built into the launch system that would prevent an inadvertent initiation of a launch event. With previous launcher prototypes, such as the RULE system, it was logical to give this "Safe to load" signal when the mechanical lock was engaged, thereby preventing launcher operation even in the event of an actuation. However, the Chain Launcher and AMPPS systems do not utilize stored energy in the same way the RULE did. Furthermore, due to the significantly lower profile and overall design of the UAV interface on the AMPPS launcher as compared to the RULE, even if the system were to actuate at the worst possible moment, that is, when a UAV is actively being loaded, the relative likelihood of damage to either the operator or the aircraft would be fairly low. Therefore, the design team determined that the system is technically "Safe to load" anytime the chain is not actively in motion, and refrained from a full implementation of this capability.

All this understood, with the successful addition of the launch-status lighting system, the underlying capabilities and software structures required to add a "Safe to load" indication

are already in place. Thus, if this functionality is ultimately determined to be desirable, a simple and easy update to the ROS *Launch* node software is all that is required.

6.4.7 Detect UAV on Launch Platform

The next capability implemented through the AMPPS design effort actually resulted in the implementation of an additional capability not selected for development as part of this process. The ability to detect UAVs loaded on the roller chain's UAV interface has several useful operational implications. First, it can be used in conjunction with the launcher status indicating lights to alert personnel in the area that a UAV is loaded on the interface. Additionally, once the ability to consistently communicate with the GCS and other UAV control stations is put into place, the launcher can provide a real-time data flow to these remote stations to ensure they are aware that a UAV is loaded and ready for launch. Furthermore, ARSENL's Zephyr II UAVs are all equipped with GoPro cameras prior to commencing flight operations to ensure full documentation of the events in each event. Currently, the launch technician has to vocally identify the date, aircraft name, and sortie number prior to initiating launches to ensure that those reviewing the footage later have a way of identifying the flight data they are observing. However, if the presence of a UAV can be detected and it can be specifically identified, perhaps an automated means of conveying this information to the GoPro camera can be developed to remove this step from the technician's launch procedure checklist.

Several methods of implementing this capability were identified through the brainstorming process: roller switches, infrared proximity sensors, magnetically-activated switches, and even laser-interrupt systems (like those used to prevent automatic garage doors from closing on people or pets) were considered. However, one method stood out as unique in its ability to both communicate the presence of a UAV as well as identify the specific aircraft on the interface. To simultaneously accomplish both the "Detect" and the "Identify" functions for a UAV on the launcher interface, a Phidgets RFID reader and some small RFID tags in the form of 15mm PVC discs were purchased.

As described in [43]:

> RFID works on the same principle as a transformer. When the reader is powered up, it gives power to a large coil. The coil creates an external magnetic

field which can then be paired with a coil inside a nearby tag. This delivers a small amount of power wirelessly to the tag. With that power, the tag is able to access a small internal memory bank and transmit a key string back to the reader via modulation on the wireless signal.

To utilize this unique technology, the Phidgets RFID reader was mounted on the underside of the roller chain support platform in the approximate area where the aircraft is affixed to the UAV interface. From here, progressive serial numbers were written to each RFID tag using a program written by the design team for this purpose. These tags were then affixed, using clear tape, to the underside of each UAV. Later, when an aircraft was placed on the UAV interface and affixed for launch, the RFID reader detected and read the data on the UAV's RFID tag. Currently, this trigger only causes the yellow light on the launcher's lighting system to actuate, notifying personnel that an aircraft has been loaded. To take full advantage of the capabilities enabled by the UAV launcher, however, a new interface is required. Thus, the AMPPS design team sourced the Phidgets liquid-crystal display (LCD) screen and control interface shown in Figure 6.5 to clearly and easily convey date, time, and UAV identification data for both the launch technician and the aircraft's onboard GoPro camera.

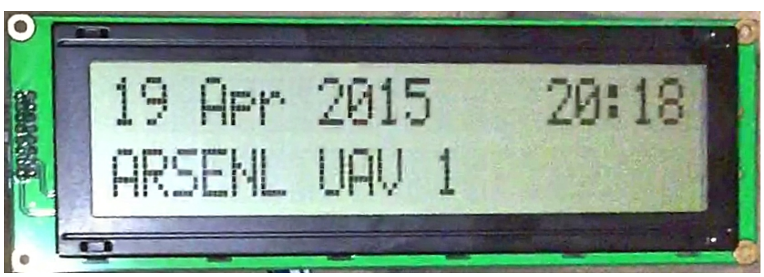

Figure 6.5: AMPPS LCD display screen

While this fully capability implementation worked initially, the LCD communication system was unfortunately unable to stand the test of time. The RFID portion of the system works flawlessly, and UAV-specific data is transmitted and used throughout the various ROS nodes when an aircraft is loaded onto the interface. However, the LCD display screen only worked for about half an hour before failing. The screen came packaged with a ten-inch, 16-wire serial cable which was not long enough to enable proper positioning of the

LCD screen on the AMPPS system. To fix this, the cable was cut in half and 18 AWG extension wires were added between all connections using plastic "butt-connectors." While this extension was successfully bench-tested and worked well upon initial assembly of the AMPPS system, one or more of the extended wire connections ultimately failed during transport to the field testing location, rendering the LCD screen useless during the first round of operational testing. To address this issue, the serial connectors on either end of the wire bundle will be disassembled, and the bundle will be re-built using longer, single-wire connections between these two connectors. This will facilitate a stronger, more reliable connection between the Phidgets LCD adapter and the actual LCD screen and will enable future users to take more complete advantage of AMPPS's capabilities.

6.4.8 Re-orient Launch System for Unfavorable Winds

The ability to re-orient the AMPPS in the event of strong crosswind or tailwind conditions is, in many respects, already implemented through the successful operation of three capabilities previously discussed. First, the ability to detect wind speed and direction is facilitated through the use of the Oregon Scientific weather station, which then transmits these parameters over the Wi-Fi network using ARSENL's custom data messaging system. Next, the launch technician is alerted to an undesirable wind status with respect to the current system orientation through a ROS node which compares the incoming wind and compass data and then actuates a flashing function for the red light on the launcher status lighting system. Finally, the operator can re-orient the system into the wind in just a matter of seconds using the wireless controller and motorized wheel systems.

While the ability to re-orient the launch system manually using the motorized wheel subsystem is currently possible, a follow-on capability that should be investigated and developed is the ability to automatically re-orient the launcher. In such a scenario, the launch technician need only press a button to initiate an automated system re-orientation. The launcher will then take a compass heading and compare this information to the latest wind data, decide which direction to turn, and then actuate the two wheel motors, turning the entire system until the compass heading matches the latest wind direction measurement. Additional sensors, such as /acGPS modules, could also be added to the system with "no-fly" zones mapped in their software, thereby ensuring safe launches even when considering obstacles which lie far out of range of the sonar safety-sensors. While this automated re-

orientation is, as of now, just a concept, the underlying data streams and software structures required to make this concept a reality are already well-established.

6.4.9 Streamline Setup and Initialization

The final capability implemented during the development of the AMPPS launch system was a more streamlined electrical and software system initialization process. For previous prototype iterations, electrical systems were powered on, and then the ROS system and all required nodes were individually initialized by the user. This process was not only slow, but it also required a moderate level of knowledge regarding the operation and use of Linux and the ROS system. For the AMPPS prototype, it was originally envisioned that the main power switch at the front of the launcher would be turned on, followed immediately by the three subsystem power switches on the rear control panel. Electrical power would immediately be provided to the two Roboteq motor controllers, as well as to the embedded computer and Phidgets sensor components, which are shown in Figure 6.6. The proprietary software internal to the Roboteq controllers would boot and prepare these systems for operation, and the Linux and ROS operating environments loaded on the ODROID single board computer would also boot and initialize all necessary Wi-Fi connections, Bluetooth connections, and ROS node executable scripts.

While this capability was implemented in part, the original vision has yet to come to full fruition. However, many of the smaller, individual pieces of this auto-boot problem have been solved, so the implementation of this full capability is close at hand. First, the ability to connect to the Wi-Fi network was hard-coded into a Linux system file that was automatically run as the operating system boots. Second, switching to the Logitech gamepad with the dedicated Bluetooth dongle enabled the system to automatically recognize the wireless controller system upon bootup. Finally, instead of manually initializing the ROS environment and all its executable nodes individually, a single *launch* file is created which starts all the required ROS scripts.

Recall that a parallel software development approach was desired for this system, where capability programming and software systems were created on both a personal laptop computer and on the embedded ODROID computer simultaneously. Since some portions the AMPPS's software systems are still in development, the launcher is currently operated

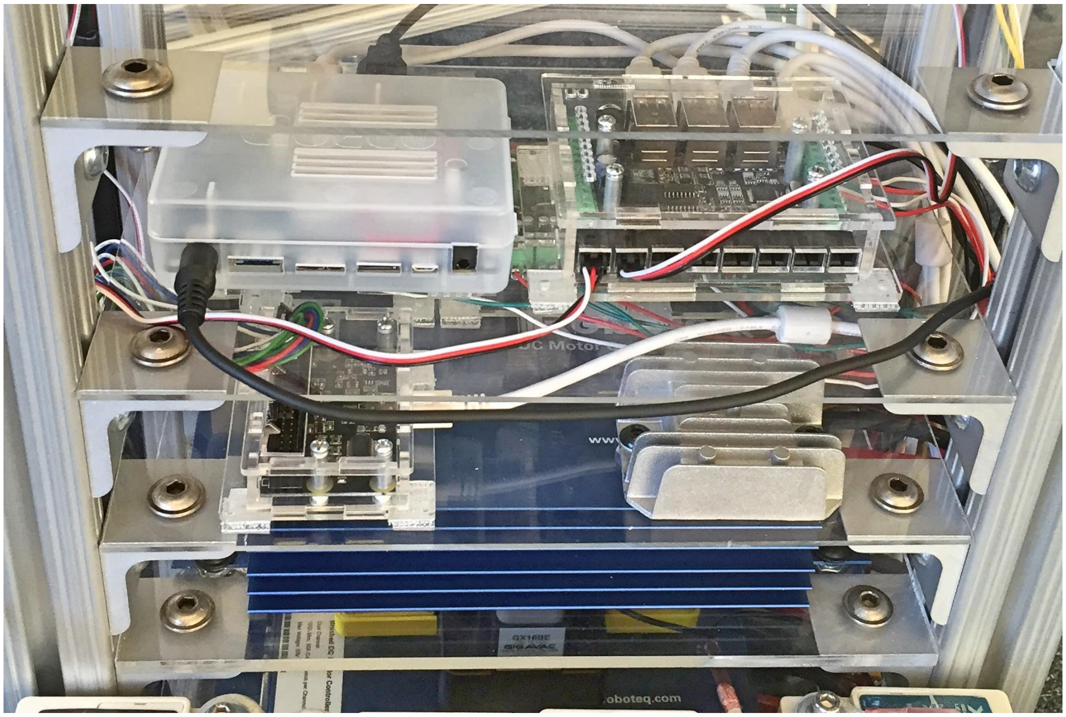

Figure 6.6: AMPPS's embedded ODROID XU computer and Phidgets Interface Kit

through the laptop computer rather than the ODROID. This enables easier adjustments to software scripts during lab and field-testing, but also necessitates a manual startup of the AMPPS's software systems. Thus, to startup the AMPPS launcher as currently configured, the operator must manually make USB connections to the laptop computer and then run the ROS `launch` file using the Linux command-line interface. However, it is expected that this procedure will soon be updated to more closely match the originally conceived startup procedures once system operation is shifted over to the ODROID computer.

6.4.10 Capabilities Not Fully Implemented

As with the previous two prototypes, several capabilities identified for development and implementation into the AMPPS launcher were either sidelined or not fully completed prior to commencing operational tests for the system. Many of these capabilities are merely disabled in software, while others will require further software or component integration efforts to ensure proper functionality.

First, the ability to disable the system's aircraft launch capability in the event that unsafe

conditions are detected (e.g., personnel standing in front of the launcher) has been implemented and tested in a lab environment, but is currently disabled to maximize system reliability. As mentioned earlier, the sonar sensors chosen for personnel and object detection are prone to periodic false-positives, causing the system to disable the launch capability at times when it should not be disabled. To address this, future development of the object-sensing capability should include the procurement and testing of multiple distance sensing devices to ensure the final launch system meets higher standards of reliability when the "disable" function is active. However, it should also be noted that further field-testing of this capability, as currently implemented, should be performed to more accurately determine the degree to which this false-positive issue actually affects the AMPPS usability in an operational environment.

Next, the ability for the AMPPS system to detect environmental conditions and to react to changes in these parameters is mostly implemented, but still requires some minor updates to software to facilitate full functionality. The weather station is set up and communicates with a nearby computer, which then transmits the key weather data parameters over the Wi-Fi network. Additionally, the AMPPS is able to connect to this same Wi-Fi network, and can receive this weather-data message. The ROS software systems are also already configured to respond to wind condition information and can detect the direction the AMPPS is pointing, resulting in an interruption to the system's ability to launch aircraft in the event of high speed crosswinds or tailwinds and the actuation of a blinking red-light on the LED tower. However, a new ROS node still needs to be written and implemented which will convert the weather-data message received over Wi-Fi to a ROS message that can be utilized in that environment. Currently, a node is hard-coded with arbitrary wind direction and speed data to enable the full testing of the ROS-side functionality without this software-based data transfer. It is worth noting, however, that the ARSENL team has already developed and tested the code required to perform this Wi-Fi to ROS message conversion and publish said message to a ROS topic. Therefore, the full implementation of this capability is likely close at hand, since all that is required is an adaptation and integration of this already-existing software.

The streamlining of the setup and computer/software initialization processes is another capability that is largely implemented, but not yet fully complete. The computer systems

are able to automatically boot, connect to Wi-Fi, and ROS and all associated nodes can all be started with a single command. However, this command is not yet automated, and the ROS-based software systems must all be started manually by the user in the current configuration. The software must also be improved to better allow for selective energization and de-energization of the Roboteq motor controllers. Currently, the mobility and launch-motor subsystems must be energized prior to booting up the computer and sensing subsystem. This ensures that the computer is able to detect and initialize software for the Roboteq controllers. However, if power is killed and subsequently re-applied to either of these motor controllers, the computer system is, as of now, unable to detect and re-initialize ROS-side software to control the systems. Thus, the majority of this capability has been successfully implemented and tested, but additional work is required to ensure the delivery of a fully robust software and electrical system.

Finally, three capabilities were not pursued in any capacity during the AMPPS development effort. As earlier discussed, launch platform position sensors were not integrated into the AMPPS since no time sensitive position-triggered functions are required for successful system operation. The system is also unable, as of yet, to communicate the status of the AMPPS's sensors and subsystems to the GCS. It is expected that this capability will be implemented soon after the development of the Wi-Fi-ROS message conversion software scripts required for the full implementation of the weather-sensing capability. Last, the ability of the launcher to receive "abort-launch" commands from either the GCS or any of the safety-observers in the vicinity was not pursued during AMPPS development. However, it is conceived that this capability could be included in future launch system iterations through the addition of a new Wi-Fi message or by incorporating a series of Bluetooth connected control devices into the launch process.

6.5 Electrical System Design

The AMPPS's electrical and sensor systems are significantly more complex than has been observed in the previous two launch system prototypes. The implementation of all the software and sensors based capabilities identified in Section 6.4 necessitated the creation of a complex wiring system in which multiple voltages and data-streams are piped to a large number of individual components. As such, for the purposes of analyzing the electrical system design for the AMPPS launcher, the overall system is broken up into three pri-

mary electrical subsystems which are reviewed individually before focusing on the fully-integrated, master electrical system.

6.5.1 Chain-launch Subsystem

The first wiring diagram created in support of the AMPPS launcher development is for the main, chain-launch subsystem. For reference, this diagram is shown in Figure 6.7.

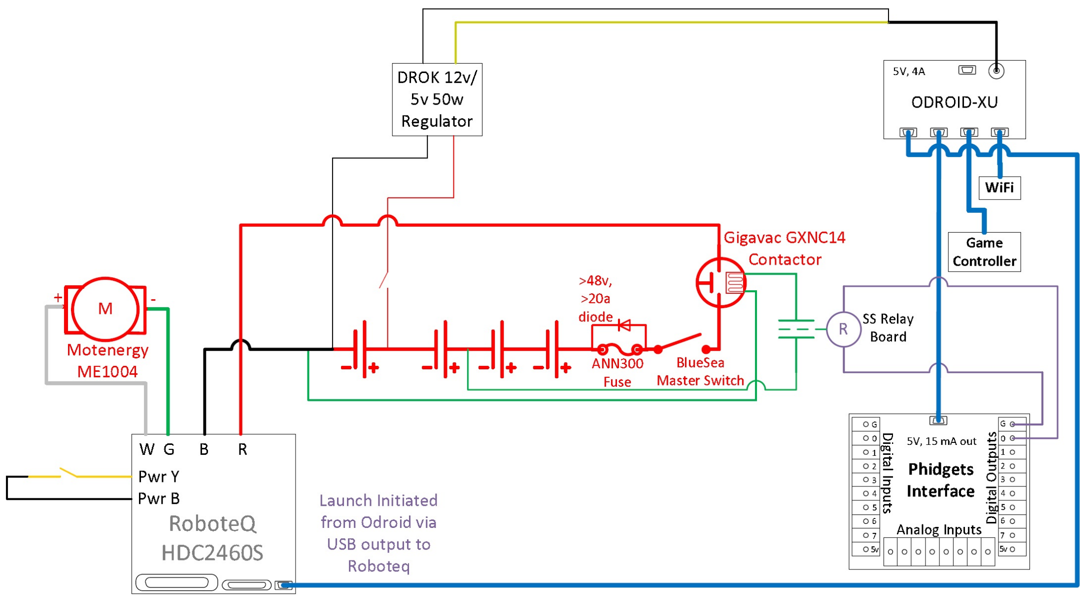

Figure 6.7: Electrical diagram for the AMPPS chain-launch subsystem

Note that this system is similar, in many respects, to the electrical diagram from the Chain Launcher prototype. First, the Logitech F710 Gamepad's Bluetooth dongle is installed in one USB port to facilitate communications between the controller and the embedded computer (or laptop). Three other USB ports are then connected to a Roboteq HDC2460S Motor Controller, a Wi-Fi USB dongle, and to a Phidgets Interface Kit using standard USB cables. The ODROID computer is powered from one of the 12 volt batteries in the main battery array via a 12-volt to 5-volt transformer and a small power switch which is run to the control panel at the rear of the launcher.

The Roboteq HDC2460S Motor Controller is connected to a Motenergy ME1004 DC motor which drives the main roller-chain and sprocket assembly, and also has a small electrical power switch that is mounted on the control panel at the rear of the launcher. It should be

noted that the Roboteq controller is designed to operate when this switch is in the "open" position, which is atypical for most electrical system wiring. When this switch is closed, the 48 volts being fed to the controller from the main battery array is tied to the controller's ground bus, thereby causing the controller to shutdown. Thus, the power switches for both this and the Roboteq HDC2450 Motor Controller used to operate the mobility subsystem are actually installed backwards in the rear control panel.

The Roboteq motor controller is also connected to the AMPPS's main power system. The ground bus in the controllers are tied to the ground terminal on one of the 12 volt lead-acid batteries. The positive terminal on this battery is then tied to the negative terminal on a second battery, and so on until the four batteries that power the system are connected in series. Current then flows through a 300 amp Bussman-style fuse, through the same master power switch used to operate the Chain Launcher prototype, and then through a normally shut Gigavac DC Contactor before being tied to the Roboteq's internal, high-voltage busses.

The operating coil for the Gigavac contactor is tied to both the system ground and to a 24 volt supply from the battery array via a Phidgets solid state relay. This relay is then connected to digital output 0 and the Interface Kit ground bus. This setup ensures that the system will power on and off as desired for normal operation, but also provides a means of issuing a remotely-triggered full-system shutdown. For this to occur, the operator calls for a shutdown via a command sequence on the Logitech gamepad. Software on the ODROID system detects this command and then sends a signal, via USB, to the Phidgets Interface Kit telling it to energize digital output 0. When energized, the Phidgets relay connected to this output is shut, providing 24 volts of DC power to the operating coil inside the Gigavac contactor. This causes the contactor to open and, in doing so, secures the 48-volt power being supplied to the Roboteq motor controllers. This results in a complete shutdown of all AMPPS systems and components other than the ODROID computer and the Phidgets Interface Kit, thereby placing it in a safe, de-energized state.

6.5.2 Mobility Subsystem

The next wiring diagram created for the AMPPS launch system, which is depicted in Figure 6.8 is to power and operate its mobility subsystem.

As with the diagram for the Chain-launch subsystem, many similarities exist between the

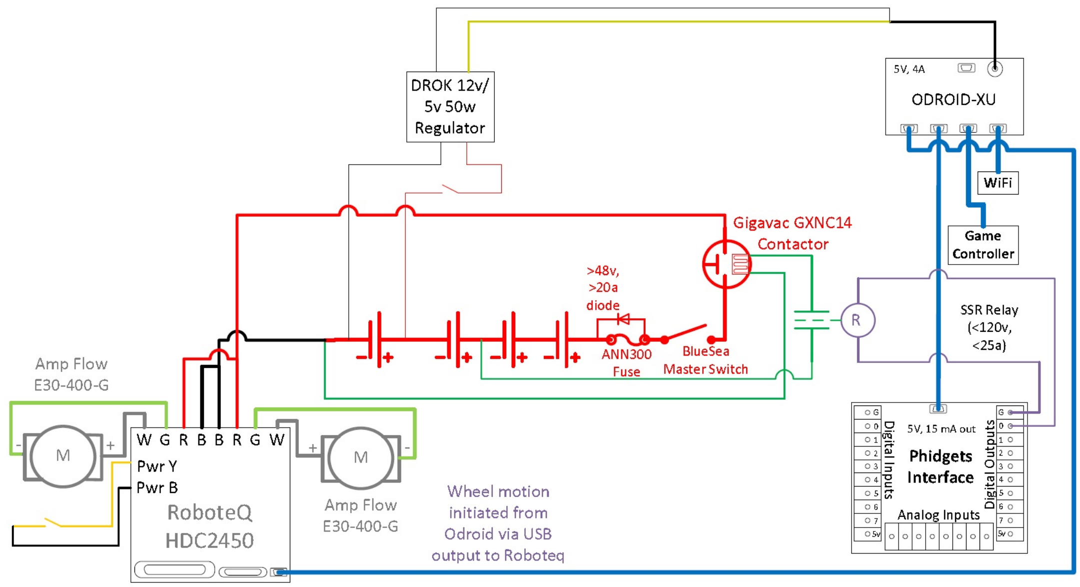

Figure 6.8: Electrical diagram for the AMPPS mobility subsystem

mobility system implemented on the Chain Launcher prototype and that created in support of the newer AMPPS system. First, the mobility system is once again controlled by the Logitech F710 gamepad which is tied to the ODROID computer (or laptop) via a Bluetooth USB dongle. Other USB ports are connected to a Wi-Fi dongle, a Phidgets Interface Kit, and to the Roboteq HDC2450 Dual-channel Motor Controller. The components involved in the power and operation of the ODROID XU computer are the same as those discussed during the Chain-launch subsystem overview and, as such, are not reiterated here.

For the mobility subsystem, an AmpFlow E30-400-G DC motor and gearbox assembly is connected to each channel of the Roboteq HDC2450 Motor Controller. As discussed in Chapter 5, the software built into the Roboteq series of motor controllers was pre-configured to enable signal mixing operations. This allows the motor controller to internally process simple "throttle" and "steering" commands from the ODROID computer system and output appropriately proportioned voltages to both motors, enabling easy and straightforward operation of the AMPPS's mobility features with little additional effort on the part of the design team.

The Roboteq motor controller used to operate the AMPPS's mobility subsystem is wired

to the main power busses and the associated electrical-safety components in the exact same manner as the controller used to operate the roller chain assembly. Finally, the Roboteq controller is also equipped with its own small, independent subsystem power switch that is mounted on the control panel at the rear of the launcher alongside the switches that control the computer and launch-chain systems.

6.5.3 Sensors and Computing Subsystems

The final electrical system diagram created during the design of the AMPPS launcher is for operating the components associated with the computing and sensing subsystems. This schematic, shown in Figure 6.9, shows all the wiring connections that are required to provide power and signal routing capabilities to all the Phidgets sensors and interfaces.

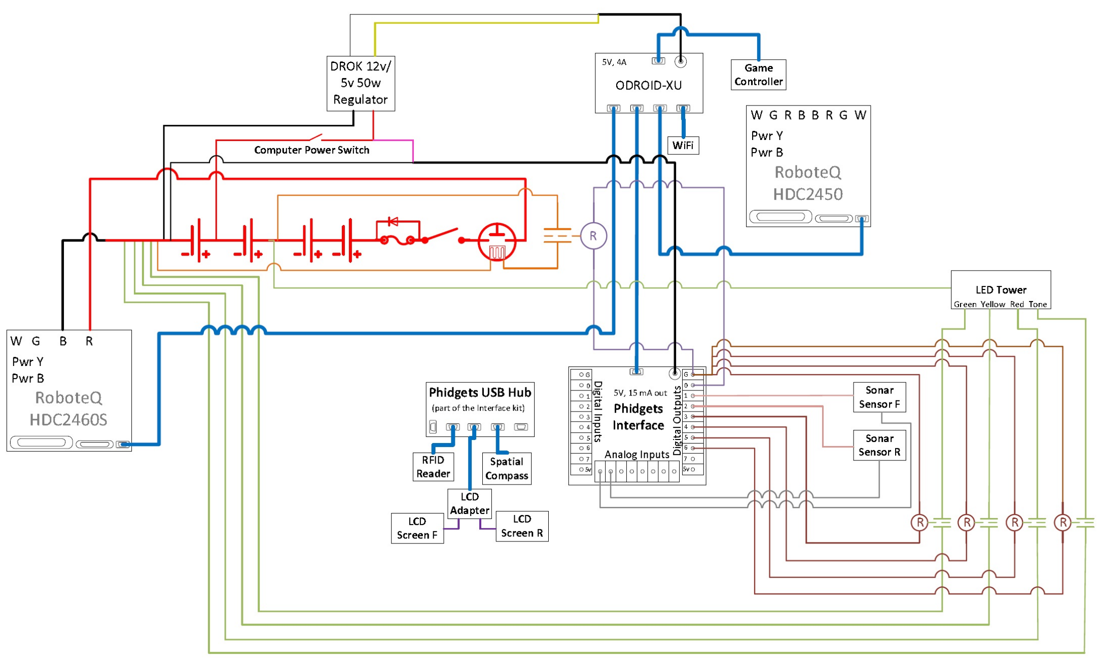

Figure 6.9: Electrical diagram for the AMPPS sensors and computing subsystems

Beginning with the ODROID XU computer (or the laptop that would be connected in its stead), note that cable-based USB connections are made to both Roboteq motor controllers and to the Phidgets Interface Kit. Additional USB ports are also allotted for the Logitech gamepad Bluetooth and Wi-Fi USB dongles. As previously mentioned, the ODROID computer is powered by one of the 12-volt lead-acid batteries in the main battery array via a

12-volt to 5-volt transformer. For safety, this 12-volt power supply is specifically provided by the first battery in the array, thereby ensuring that the reference ground for all systems remains consistent. Also powered from this battery is the Phidgets Interface Kit and its onboard, six-port USB hub which is used to connect several of the Phidgets USB-based sensors and components. Power to both the ODROID and the Interface Kit is controlled via a single switch which is located on the control panel at the rear of the AMPPS.

Next to the Phidgets 1019_1 Interface Kit, at the center of the diagram, is its powered, 6-port USB hub. This hub provides for the connection of the Phidgets RFID reader, the Phidgets spatial compass, and the Phidgets LCD adapter.

The Phidgets Interface Kit's analog input ports zero and one are connected to the two Phidgets sonar sensors mounted at the front and the rear of the AMPPS. These sensors are also connected to the Interface Kit's digital output ports one and two, allowing for the selective energization of the two sensors. Other digital outputs, connected to ports three through six, are connected to the relays which operate the red, yellow, and green lights, as well as the auditory tone on the LED lighting tower. A final digital output port on the Interface Kit is connected to a fifth relay, which controls the actuation and opening of the Gigavac DC contactor.

The final portion of this electrical diagram that merits discussion is the main system power loop and all the connections made to the various junctures. Beginning with the main system ground bus, major connections are made to both the first 12 volt lead-acid battery and to the Roboteq motor controllers' internal ground buses. Also connected to this ground bus are connections to the four relays which control the lights and tone on the LED tower, the ground wire for the Phidgets Interface Kit DC power plug, the ground wire from the Gigavac contactor operating coil, and a ground wire connection to the 12-volt to 5-volt transformer. As mentioned previously, a connection to the computing and sensing system's power switch is made to the positive terminal of the first 12-volt battery in the battery array. Additional connections to the battery array, at the positive terminal of the second 12-volt battery, are made to the relay which operates the Gigavac contactor and to the main power lead for the LED lighting tower.

6.5.4 Integrated Electrical System

After creating the wiring diagrams for each of the AMPPS's subsystems, a final master electrical schematic is created which integrates the three subsystem diagrams into a single drawing. A thorough overview of all the connections in this diagram, which is shown in Figure 6.10, has already been detailed in the preceding sections of this chapter and, as such, will not be further expounded upon here. However, there are other intricacies of the AMPPS's electrical systems that still merit discussion.

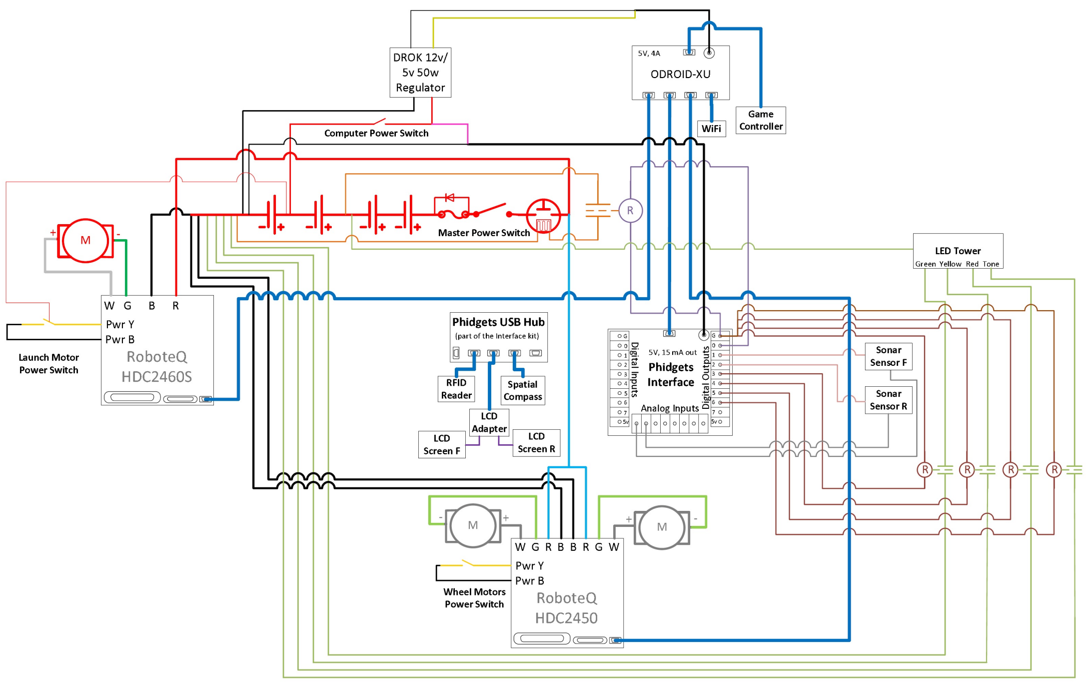

Figure 6.10: Master electrical diagram for the AMPPS

First, as with the Chain Launcher prototype, it should be noted that the DC motors attached to the wheels and the roller chain assembly are capable of drawing transient DC currents in excess of 200 amps. However, most of the Phidgets sensors and computing system components require only minimal current flows to ensure consistent and reliable operation. Thus, proper wire sizing decisions are key to the safe construction of the AMPPS electrical systems. As such, all the wires in these diagrams connected to the main-power loop, which begin with the battery array and terminate with connections to the two Roboteq motor controllers and their associated DC motors, are sized to be eight AWG or thicker. Conversely, due to their lower power requirements, all wires connecting the computing components or

Phidgets sensors are sized to be either 18 or 20 AWG.

Several additional electrical safety considerations are made during the construction of the AMPPS launcher. First, a Bussman-style 300 amp fuse is connected in series with the two Roboteq motor controllers to ensure these critical components are protected from a potential overcurrent condition. Next, electrical bus bars with protecting covers and wiring connection posts are used to facilitate the various connections to the electrical system ground and 24-volt junctures. All electrical components are also mounted to clear acrylic cases and shelves, as shown in Figure 6.11.

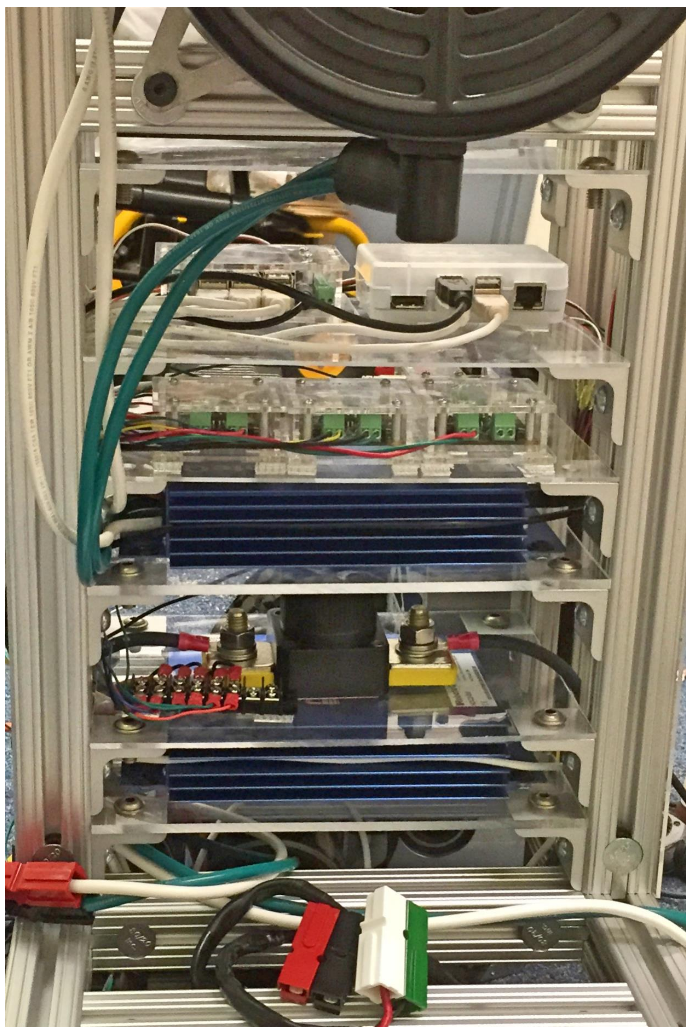

Figure 6.11: AMPPS electrical component shelving system

The entire electrical component shelving assembly is encased in a second clear-acrylic enclosure to protect against incidental personnel contact with electrical system components and to provide a degree of environmental protection for the key electrical components built into the system. The use of the clear acrylic for these mounting and component protection functions also helps facilitate an easy diagnosis of electrical problems in the field, should they arise. Further electrical safety measures include the use of crimped and insulated ring terminals to make the majority of the connections between electrical components and the use of insulating rubber or plastic boots to protect otherwise-exposed connections to the terminals on the batteries and DC motors. All wires outside the electrical component shelving assembly are also, to the maximum extent possible, routed inside the extruded aluminum channels and have a plastic channel cover that minimizes the potential for loose or stray wires which could snag on external components during transport.

Finally, as previously discussed, there are multiple means of de-energizing the AMPPS on various system levels. The master power switch and normally shut Gigavac DC contactor provide a physical and software-based means of initiating a full system shutdown. There are also individual subsystem power switches routed to the electrical control panel at the rear of the AMPPS which facilitates a de-energization of each subsystem individually. The use of these multiple power switches provides redundancy since multiple switches must be flipped to provide power to the systems, and also provides multiple physical and software-based means of placing the AMPPS systems in a safe, de-energized condition.

6.6 Software System Design

As with the previous two prototypes, the software which drives the operation of the AMPPS is written using the Python language to operate in the ROS environment. However, as the AMPPS utilizes significantly more sensors and components than either of the previous two launch systems, the number of nodes written and incorporated into the newest software architecture is also increased. For reference, the ROS communications diagram which shows the originally projected, full range of AMPPS system functionality is shown in Figure 6.12.

In the diagram, all hardware-based components are shown at the top in light blue boxes. A walkthrough of the functionality of this proposed software system begins, as before, with the Logitech F710 Gamepad that is pictured at the center of the diagram. This wireless con-

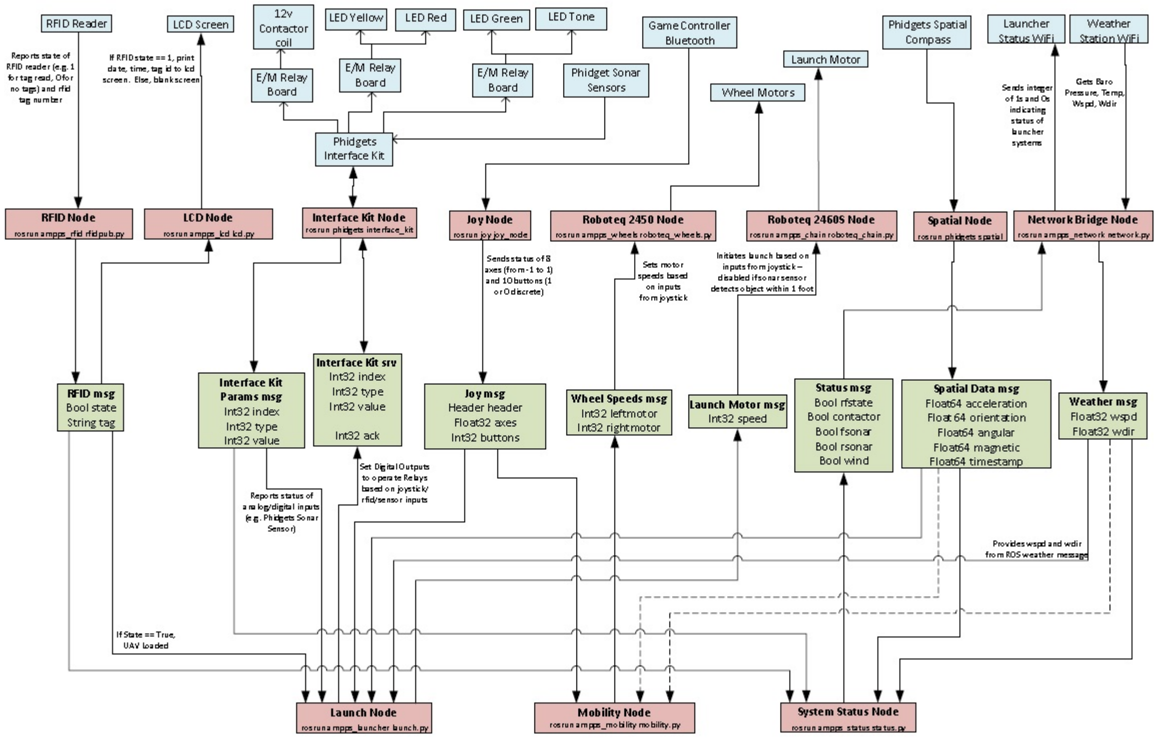

Figure 6.12: ROS communications diagram showing targeted range of funtionality for the AMPPS launcher

troller communicates with the ODROID or laptop computer over a Bluetooth connection to the corresponding USB dongle. The ROS *Joy* node contains scripts and drivers that detect inputs from the gamepad and publishes the states of each button or joystick to the *Joy* topic in the ROS environment. As with the Chain Launcher prototype, this *Joy* topic is subscribed to by both the *Launch* node and the *Mobility* node, which then issue commands to drive their corresponding Roboteq motor controllers via ROS messages published to the *Launch Motor* and *Wheel Speeds* topics. The *Roboteq HDC2460S* node and *RoboteqHDC2450* node then subscribe to these topics and, when messages are published, these nodes use a suite of open-source Roboteq driver scripts to issue speed commands to the motor controllers. As with the Chain Launcher prototype, most of the software used to create and operate these *Roboteq* nodes was written by another member of the open-source community. This individual originally adapted the Roboteq drivers and publicly-available API for use as a ROS-compatible node in the open-source, downloadable `ros-roboteq-hdc2450` package. Minor modifications to the scripts in this package then enable AMPPS to actively

communicate with both the motor controllers at the same time using the ROS interface.

In addition to the ROS *Joy* topic, the *Launch* node subscribes to several other topics in the ROS environment that should be highlighted. First is the *RFID* topic, which contains two variables corresponding to the RFID reader's tag-detection status and the serial number associated with any tag being detected by the reader. The *RFID* topic is published by the *RFID* node, which also provides the software required to communicate with the USB-connected Phidgets RFID reader. To display the information being read by the RFID reader to the launch technician, the *RFID* topic is also subscribed to by the *LCD* node which, in turn, drives the LCD screen via a wired connection.

Next, the *Launch* node subscribes to the *Interface Kit* params topic to monitor the status of the two Phidgets sonar sensors which are connected to the kit's analog input ports. The *Launch* node also publishes to the *Interface Kit* service call, enabling an actuation of the various Phidgets relays that operate the LED lights, signaling tone, and the Gigavac contactor operating coil through the *Interface Kit* node.

The *Launch* node also subscribes to both the *Spatial Data* topic and the *Weather* topic. Using the data from these two topics, the computer performs a comparison of the AMPPS's magnetic heading and the current wind direction and, if a significant difference exists, the ability to launch an aircraft is disabled. As alluded to previously, the *Weather* topic in the diagram is published by a *Network Bridge* node, which receives the weather data message being transmitted over the Wi-Fi network and converts it to a ROS message for use by other systems. Similarly, the *Spatial Data* topic is published by the *Spatial* node, which is written to take the raw data from the Phidgets Spatial 3/3/3 device and convert the information into a 360-degree heading that can be utilized by other portions of the AMPPS's software system.

Finally, in addition to the computationally heavy *Launch* and *Mobility* nodes, a new computational node is proposed for the AMPPS. This *System Status* node, subscribes to the data published to the *RFID*, *Interface Kit Params*, *Spatial Data* and *Weather* topics in ROS and, essentially, concatenates the data from these streams into a set of boolean values. These values are intended to eventually communicate the status of various software-side sensor statuses, such as the detection of an RFID tag or whether wind conditions are consistent

with the launcher's current orientation, to the GCS. The *Status* topic is subsequently subscribed to by the *Network Bridge* node, which converts the ROS *Status* messages to strings of boolean values which can be transmitted over the Wi-Fi network using a new message component. This operation would provide the GCS, as well as other stations connected to the Wi-Fi network, the ability to monitor conditions on the launcher in real-time with minimal vocal or human-to-human interaction required.

Unfortunately, as discussed previously in Section 6.4, not all portions of the originally projected AMPPS software system were fully implemented in the initially tested configuration. Recognizing this, the software communication system that is actually implemented is shown in Figure 6.13.

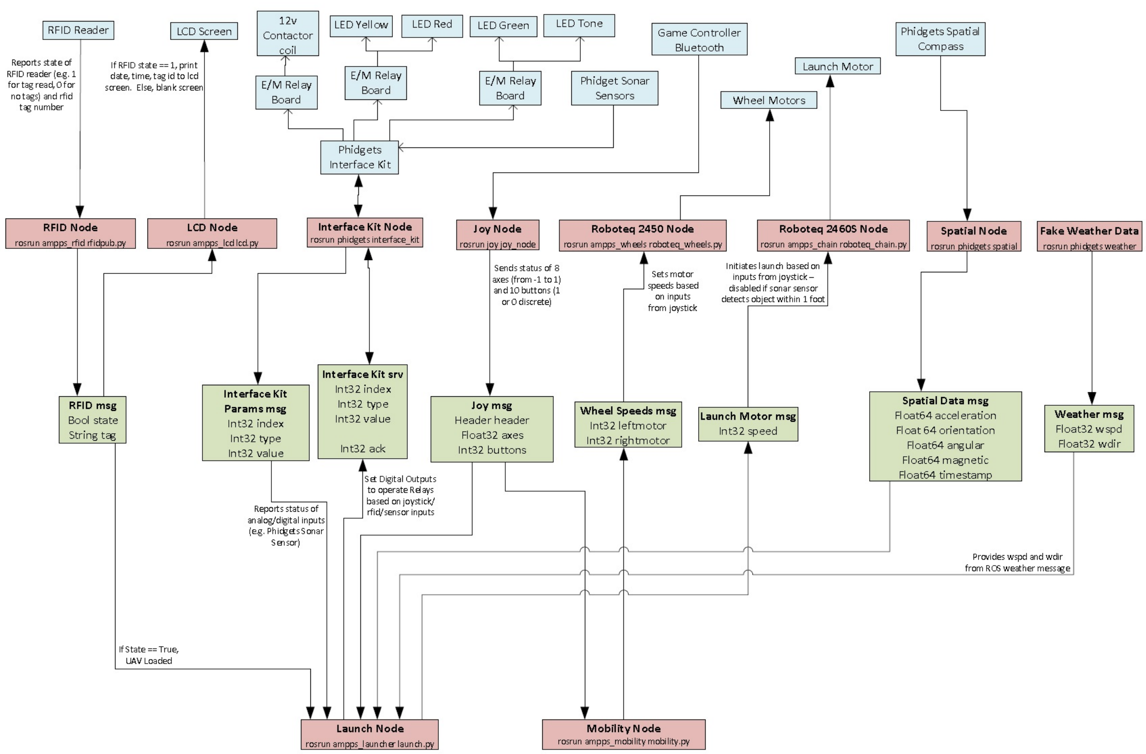

Figure 6.13: ROS communications diagram showing delivered range of functionality for the AMPPS launcher

The majority of the functionality summarized for the first ROS communications diagram is relatively unchanged in the actual AMPPS system implementation, although some key differences do exist. These differences primarily stem from the current lack of an opera-

tional *Network Bridge* node which convert Wi-Fi messages for publication to ROS topics and, conversely, convert ROS messages back into Wi-Fi data structures for transmission over the network. As a result, a fake *Weather Data* node is currently used to publish an arbitrary, hard-coded set of wind speed and wind direction parameters to the *Weather* topic. This allows other portions of the software system, which utilize the wind, speed, and heading data, to be tested with these weather-based operations largely in place. Finally, both the *System Status* node and the *Status* topic are both removed from the current AMPPS software system since there is currently no means of transmitting the information provided by these structures to external operating stations.

6.7 System Testing and Conclusions

Due to severe time constraints towards the end of the launcher-development effort, the AMPPS prototype was actually fully constructed, wired, and field-tested in a span of only six days. However, despite this extremely short turnaround-time, the AMPPS launcher is generally considered to be a highly successful rapid launch-system prototype. Shown in Figure 6.14, the AMPPS meets or exceeds nearly all the initial requirements set-forth for the UAV launch system and also showcases a wide range of technologies and capabilities that are unique to this particular application.

Ultimately, the AMPPS was able to support aircraft attachment to the roller chain, detect and identify the particular aircraft attached to the chain, detect conditions in the surrounding area and communicate anomalies to the launch technician, wirelessly receive operating commands from the launch technician, provide a visual and auditory countdown for launch, accelerate the attached aircraft in a highly controlled launch sequence, release the aircraft without causing any significant damage, and then reset the attachment interface with no (or minimal) user input. The system is also capable of maneuvering over paved and offroad terrain, can be quickly and easily shutdown via software or through physical manipulation of power switches, and can also be powered-on and started up in only a matter of minutes.

The AMPPS executed more than twenty successful launches of UAVs utilizing automated propulsion actuating systems during its first series of operational tests. While some rapid-launch system requirements went untested, such as the actual time required to get 50 UAVs airborne during a single swarm-launch event, most of the untested requirements can be

Figure 6.14: AMPPS demonstrating a successful UAV launch

extrapolated from existing data. Based on videos and observations made during initial operational testing, the AMPPS is capable of executing a launch event at least once every 12 seconds. This metric can also likely be further reduced by reprogramming the Roboteq HDC2460S controller to decelerate the chain more quickly following launches and through the pre-staging of aircraft close to the launcher's vicinity for easier retrieval and faster loading by the launch technician.

While most aspects and functions of the AMPPS were successful, the system is, of course, not without its drawbacks. Mechanically speaking, the second aircraft launched by the

AMPPS during field-trials was damaged by the UAV attachment mechanism, prompting an overnight attachment system redesign that, ultimately, proved to be a highly successful and well-received alternative. Several software and electrical issues also emerged throughout testing. As mentioned previously in Section 6.4, the wire bundle which connects AMPPS's LCD screen to the Phidgets LCD adapter was too short and had to be extended. While initial bench tests and subsequent AMPPS laboratory testing with the LED systems fully integrated was conducted, the screen proved to be non-functional during operational field tests. Since no issues were found in software for this capability, it is assumed that the problem can be attributed to this modified wire bundle.

Another issue with the AMPPS launcher, as originally tested, is the sensitivity of the wheel motors to low-level joystick throttle and steering commands from the wireless Logitech gamepad. The controls for the launcher perform exceptionally well when it is traveling at higher speeds, but fine-tuned, slow speed control of the wheel motors is difficult to achieve. While it is expected that this problem can easily be solved by using software to create a more exponential joystick control structure, the issue currently remains unaddressed.

One more significant software-based issue for the AMPPS launch system is its inability to automatically regain connection with the Roboteq motor controllers after lost connection occurrences. Currently, when power is secured to only one of the Roboteq controllers, the software system detects the lost connection but is unable to re-establish that connection when power is restored to the device. Instead, when commands are issued to the previously-secured device, error messages are displayed which eventually overwhelm the system and necessitate a full shutdown and reboot of all three AMPPS subsystems. As such, further development of the software for the AMPPS is currently required to address this problem and promote maximum software reliability.

Finally, the AMPPS will still benefit greatly from the full implementation of all the capabilities which were only partially-developed through the AMPPS development effort. A full shift to the ODROID computer is needed, which also requires the full implementation of the software auto-initialization capability. The scripts for the *Network Bridge* ROS node need to be completed, thereby facilitating the real-time flow of wind data from the weather station to the launcher via the Wi-Fi network. This node will also enable the launch system to transmit the status of its own systems out over the network for use by other operating

stations involved in the swarm effort. Ultimately, it is hoped that many of these capabilities can be developed and implemented into the AMPPS system prior to ARSENL's next multi-aircraft field-testing event, facilitating an even more capable system that can help the team meet its swarming aircraft goals both near-term and into the future.

CHAPTER 7: Conclusions and Recommendations

7.1 Summary

The primary goal for the work completed in support of this project was to develop a launch system for fixed-wing UAVs that was easily transportable, straightforward to operate, and was capable of very short launch-cycle times. More specific to this particular research effort was the identification, prioritization, selection, and implementation of enabling electrical, software, and sensors-based capabilities that led to increases in the launch system's efficiency, usability, and margin to operator safety. Ultimately, the launch system design team was able to develop a set of prototypes that exhibited varying, yet generally expanding degrees of capability, culminating in the creation of a revolutionary launch-system for fixed-wing UAVs.

The prototyping process began through the development of a clear understanding of the context and environment in which the new launch system was expected to operate. This enabled the launcher design team to more clearly determine and articulate system requirements and performance parameters. Next, a spanning set of likely operational scenarios were defined and, from these scenarios, a comprehensive list of potential launch-system capabilities were identified. Capability priority metrics were then established to help facilitate the prioritization and organization of these potential capabilities. For this effort, the three metrics selected were the number of operational scenarios to which the capabilities would likely contribute, an overall estimate of the utility provided by the capability to the launch process, and an estimated degree of difficulty associated with the implementation of each capability. Nominal, as well as maximum and minimum values were then assigned to each individual capability for each metric. Then, using these nominal and maximum and minimum scores, an Analytic Hierarchy Process analysis was performed. This process was also executed several more times, using the minimum and maximum values identified for each metric both in isolation and in concert with each other. Eventually, an average AHP score was assigned to each metric, and all the capabilities were ranked based on these scores. Finally, natural gaps in the capability scores were identified, and groups of potential

capabilities were designated for implementation into the various launch-system prototypes.

The first launch system prototype was the Rapid UAV Launch Engine, which utilized a tank of compressed air, a solenoid-operated, three-way pneumatic valve, and a pneumatic actuating cylinder as the means of propelling a UAV mounted at the opposite end of a lever arm and pivot assembly. The system also leveraged a laptop computer running Linux and the Robot Operating System to control software-side functions and facilitate more efficient system operation. Enabling capabilities originally identified for implementation into this prototype were:

1. Abort launch functionality
2. Mechanical-based kill switches with easy accessibility
3. Moved and setup by one to two technicians
4. Lighting system to warn personnel of launch status
5. Automatic reset capability
6. Launch platform position sensors

Unfortunately, the RULE fell short in its primary directive: launching UAVs at speeds sufficient to sustain temporary flight. The RULE also suffered from other issues, such as poor reliability during full speed operation, poor mobility during operation, and poor system range due to AC power requirements. However, the system did succeed in demonstrating, at a low level, the value that an automatic reset capability could provide to an operator engaging in rapid UAV launching operations.

The second prototype system was the Chain Launcher, which used a bank of four lead-acid batteries and, eventually, a DC motor controller to provide power to a large DC motor connected to a roller chain and sprocket assembly. Once again, the prototype's functionality was controlled through a laptop computer, a wireless game controller, and several USB connections to the key system components. Enabling capabilities identified for implementation into this second prototype were:

1. First-level capabilities not fully implemented or requiring significant design changes from first prototype
2. Safe to load indication

3. Detect environmental parameters (wind data)
4. Maximize launcher range envelope from ground control station
5. Detect people/objects in launcher vicinity
6. Detect UAV on launch platform
7. Disable launch capability if winds averse

In most respects, the Chain Launcher design was considered to be a success. It was able to be easily setup, was capable of accelerating and releasing ARSENL's UAV at the desired launch speed, and was able to be configured for an automated reset through the use of software and precisely timed motor commands. The Chain Launcher's design also necessitated an emergent requirement for a powered-wheel subsystem with a wireless external controlling device. However, even with these new capabilities successfully integrated into the design, the system was still not all that a UAV launcher should be. It was hastily built, hastily wired, and lacked many of the enabling capabilities that should theoretically have been implemented and included in this prototype iteration.

Finally, development work commenced on the final Automated Multi-Plane Propulsion System. This prototype was functionally similar to the Chain Launcher that came before, but included a number of refinements and additional capabilities that would have been nearly impossible to incorporate into the Chain Launcher's design configuration. The AMPPS launcher also used motor controllers to operate the chain-drive and wheel motors, and was configured for wireless, stand-off control-ability. Enabling capabilities identified for implementation into the final AMPPS prototype were:

1. First and second-level capabilities not fully implemented in first or second prototypes
2. Disable launch ability if area unsafe
3. Streamline setup and initialization
4. Communicate launch system/sensor status to GCS
5. Receive "Halt Launch" commands from GCS or safety observers
6. Re-orient launcher if wind direction not favorable
7. Disable launch ability until UAV loaded

Ultimately, the overall design and implementation of the AMPPS launch system was considered to be a resounding success. It was even easier to set up than the Chain Launcher,

provided for standoff mobility using the powered wheels and wireless gamepad interface, controlled the acceleration profile of launched aircraft to with unheard-of accuracy and consistency, and was capable of automated reset through software-based timing functions. The AMPPS also alerted the operator of personnel in the launch path, wind conditions inconsistent with the launcher's orientation, and could automatically identify the specific aircraft loaded onto the launcher interface and communicate this information to the launch technician or onboard camera systems. In field testing, the system executed more than twenty successful launches, with only one anomalous launch attributed to a flaw in the specific aircraft's construction. This anomalous launch led to an overnight re-design of the roller chain-UAV interface, and the new interface was successful in facilitating the remaining launches with no noteworthy issues. In general, testing results for the AMPPS indicated a generally well-designed and well-constructed rapid-launch system with significant potential for getting large numbers of UAVs in the air.

Finally, the financial costs associated with the development of these launch-system prototypes were not identified as a key design priority by the project stakeholders. However, the topic does merit a brief discussion. Overall, the design teams spent an estimated total of $18,000 on the development of these three prototype systems. The cost of parts and components actually installed in the final, delivered AMPPS prototype was approximately half this cost, at $9,800. For reference, a full, itemized parts list identifying all components that were procured and actually installed into the final prototype is shown in the appendix.

7.2 Recommendations

While UAVs and many of their supporting technologies have been around for a couple of decades now, the development of UAV launch systems, and especially those with rapid-fire capabilities, remains a relatively new and emergent field of study. As such, there are numerous immediately available opportunities whose study would help expand this new body of research and could help further the utility of the AMPPS system itself. Several such areas might include:

Expanded use of LCD screens: The AMPPS launcher is currently only equipped with a single LCD screen unit. This screen is normally blank, but displays date, time, and aircraft identification information when a UAV is detected on the launcher interface. However,

it was originally designed to incorporate two screens that are controlled independently of one another and would likely provide more continuous displays of information. With only a basic understanding of the ROS system and the Python programming language, one could easily develop and implement myriad new displays and data flows to these screens, providing another means of communicating the status of various launch system parameters to the launch technician.

Automated re-orientation based on wind changes: The development of the AMPPS system saw the implementation of two key capabilities that facilitate the emergence of a new potential capability. AMPPS's ability to detect and interpret wind conditions relative to its own orientation and its ability to move using a motorized wheel subsystem opens up the possibility for automated re-positioning due to un-cooperating wind conditions. In such a scenario, the launch system's computer will notify the operator of its desire to re-orient itself via either the lighting tower or the LED screen system. The operator will then press a simple button combination on the Wi-Fi controller to trigger the system to auto-position. The launcher's computer would then send speed commands to the Roboteq motor controller driving the wheel motors, resulting in a system re-orientation. Once the system's position is consistent with the direction of winds, the wheels would stop and the process would secure until a new request is initiated by the computer. While more in-depth than the previous recommendation, this capability would be significantly useful and can still be implemented in a matter of weeks with only a limited background in software programming.

Bluetooth or Wi-Fi based remotes with process "kill" switches for safety observers: To further enhance the margin to personnel safety for the AMPPS launch system, it would be useful if multiple operating stations had the ability to pause or prevent a launch-event based on the simple push of a button. Conceptually, each player involved in a swarm-launch event will have their own small remote assigned that is mounted on a belt or placed in a pocket for easy access. Each of these remotes are setup to send a command to the launcher's computer system via either a Bluetooth or Wi-Fi dongle which changes the state of a boolean variable, thereby preventing the chain-drive assembly's ability to actuate. This particular capability implementation would be both software and hardware based, and would therefore require an understanding of ROS-based software systems and and ability to design, procure, or adapt hardware interfaces for this purpose.

7.3 Future Work

While the implementation of the capabilities described in the previous section offer a lower-level ability to get involved in the UAV launch-system development process, other opportunities exist for those who desire a more substantial challenge in terms of design ideation, capability implementation, and system integration. Several such areas for future research are:

Full integration with GCS and Swarm Commander computer interfaces: One area of research with immediate implications for the operation of the AMPPS launcher is the full integration of the system's sensing, computing, and communication systems with external ground control stations. This theoretically involves the design of applications, windows, or control bars that can provide the GCS operator with a real-time status of the launcher's key operating parameters. A fundamental piece of this capability, the *System Status* node in the Robot Operating System, has already been identified and partially implemented through the AMPPS development. However, parameters communicated through the current version of this node may not represent the full range of useful data that might be transmitted to outside operating stations. Additionally, work is needed to identify and define the best way to make this data readily available and useful on these other computer systems. Those desiring to pursue such a course should have a background software application, computer interface, or software simulation system design and would likely need a strong grasp of multiple programming languages.

Multiple, interconnected launch systems: As alluded to in the third operational scenario defined in Chapter 3 of this work, the execution of a full, 50 versus 50 UAV air war will require more than just a single launch system. While the Automated Multi-Plane Propulsion System provides an excellent baseline for the development of future UAV launch systems, the additional capability for separate launch systems to communicate and de-conflict their statuses with one another with little or no operator involvement might prove to be integral to getting larger numbers of aircraft airborne in a short period. As such, this effort would first involve the construction of a second launch system that is, preferably, similar to the AMPPS system in most respects. The bulk of the unique research for this effort, however, will involve the development and testing of wireless-based communication systems between the two launchers, enabling new automated control functions built on these archi-

tectures. Pursuit of this work would require a background in both software and electrical system design, and a thorough understanding of mechanical systems would also be helpful.

Design, development, and testing of an aircraft hopper which auto-loads UAVs onto a launcher's interface: This final area for future development is much more mechanically inclined then the previous two, although there is certainly potential for software, computer, and electrical system integration efforts to automate certain processes or functions. Ultimately, it would be immensely useful to the ARSENL team to have a UAV hopper which is pre-loaded with flight-ready aircraft. This hopper will sit over top of or interface directly with ARSENL's primary launch system, and will load aircraft onto the launcher's UAV interface either automatically, as part of a consistently-timed launch sequence, or based on a physical input from the launch technician. As with the development of the launch system described in this work, this is a problem that primarily requires a mechanical solution, but the number and complexity of additional, enabling capabilities that could be built into the system are limited only by the designer's imagination. An additional challenge associated with the construction of this interface is the ability to ensure that all UAVs in the hopper remain in sync with GPS satellites while they are stacked, staged, and waiting for launch. Ultimately, the design and construction of a functional UAV hopper for the ARSENL team will require an individual with a strong background in mechanical systems, but the effort can also be adapted for those whose interests lie more in the electrical engineering or computer science fields.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX: AMPPS Itemized Budget

Table 1: Itemized Budget for AMPPS Mechanical Components

Item	Vendor	Description	Part Number	Unit Cost	Quantity	Total Cost
1/2" Shaft Base Mount	McMaster	1/2" Shaft Base Mount	185K3	$17.40	2	$34.80
1/2" Steel Drive Shaft	McMaster	1/2" Diamter 36" Long, Steel Shaft	1346K19	$23.53	1	$23.53
Collar Clamp	McMaster	1/2" Diamter Shaft Clamp, One Piece	6435K14	$2.11	4	$8.44
1/4" Key Stock	McMaster	Spring Steel Standard Key Stock, 1/4" X 1/4", 36" Length	98535A450	$11.10	2	$22.20
ANSI 40 Idler Sprockets	McMaster	Steel Idler Sprocket for ANSI Roller Chain, Low-Profile Hub, for #40 Chain, 1/2" Pitch, 1/2" Bore	6663K41	$28.78	2	$57.56
ANSI 40 Roller Chain	McMaster	Roller Chain, ANSI No. 40, 1/2" Pitch, 20' Long	6261K173	$90.80	1	$90.80
Connecting link for ANSI No. 35 Roller Chain	McMaster	Connecting Link for ANSI No. 35, Roller Chain	6261K191	$0.82	2	$1.64
Connecting link for ANSI No. 40 Roller Chain	McMaster	Connecting Link for ANSI No. 40, Roller Chain	6261K193	$0.87	2	$1.74
Horizontal tab attachment link for ANSI 40	McMaster	Roller Chain Attachment Link, Connecting Link, K-1 Tab Style for ANSI #40 Chain	7321K7	$2.94	3	$8.82
Shoulder Screw for Japanese Bearings	McMaster	Alloy Steel Shoulder Screw, 1/2" Diameter x 5/8" Long Shoulder, 3/8"-16 Thread	91259A709	$1.97	8	$15.76
Mount Tabs for Wheel Motors	McMaster	Concealed Connector, 1/4" Thread Size for 1" HT, Aluminum T-Slotted Framing Extrusion	47065T155	$1.79	8	$14.32
Wheel motor mount bolts	McMaster	Drop-in Fastener with Stud, 5/16"-18 Thread Size for, Aluminum T-Slotted Framing Extrusion	47065T234	$1.58	4	$6.32
Extra Concealed Fasteners	McMaster	Concealed Connector, 5/16" Thread Size for 1-1/2", Aluminum T-Slotted Framing Extrusion	47065T156	1.91	12	$22.92
Switch Panel	McMaster	Optically Clear Cast Acrylic Sheet, 1/8" Thick, 12" X 12"	8560K239	8.63	1	$8.63
Extra Anchor Fasteners	McMaster	Adjustable Connector, 5/16" Thread Size for 1-1/2" HT, Aluminum T-Slotted Framing Extrusion	47065T154	3.89	10	$38.90
Sheet for linear guide	McMaster	UV-Resistant Clear Extruded Acrylic Sheet, 3/16" Thick, 24" X 48" Sheet	8589K64	$42.50	1	$42.50
Linear Guide	McMaster	Roller Chain Guide, Center Channel with Walls for ANSI #40/2040, 4' LG	93095K5	$30.96	3	$92.88
Shelving Supports	McMaster	Aluminum T-Slotted Framing Extrusion, 90 Degree Bracket, Single, 2-Hole, for 1-1/2" Extrusion	47065T224	$4.06	24	$97.44
Screws to mount Shelves	McMaster	Flanged Button-Head Socket Cap Screw, 316 Stainless Steel, 5/16"-18 Thread, 3/4" Long, Packs of 10	90909A532	$11.69	3	$35.07
Nuts for shelf screws	McMaster	ASTM F594 Type 18-8 Stainless Steel Hex Nut, 5/16"-18 Thread Size, 1/2" Wide, 17/64" High, 50 pack	92673A119	$5.86	1	$5.86
Spare Nuts	McMaster	ASTM F594 Type 18-8 Stainless Steel Hex Nut, 1/4"-20 Thread Size, 7/16" Wide, 7/32" High, 50 pack	92673A113	$3.79	1	$3.79
U-Bolt Guard	McMaster	Zinc Plated Steel U-Bolt, 1/2"-13 Thread Size, 8 3/4" ID, 10 3/8" Height	3043T4	$6.97	1	$6.97

continued ...

. . . Table 1 continued

Item	Vendor	Description	Part Number	Unit Cost	Quantity	Total Cost
Primary Sprocket	McMaster	Finished-Bore Sprocket with Hardened Teeth, for #40 Chain, 1/2" Pitch, 16 Teeth, 1" Bore	2500T48	$22.34	1	$22.34
80/20 Frame	GA Worth Company	Aluminum Framing		$1,187.59	1	$1,187.59
7" Main Drive Sprockets	McMaster	Finished-Bore Sprocket for ANSI Roller Chain for #40 Chain, 1/2" Pitch, 42 Teeth, 1" Bore	6236K14	$60.50	2	$121.00
Pneumatic Caster Wheel	Uline	Pneumatic Caster - 8 x 2 1/2", Black, Swivel with Brake	H-3328BL-SWB	$49.00	2	$98.00
Motor mount and pillow bearing mounting screws	McMaster	Alloy Steel Shoulder Screw 3/8" Dia X 1/2" Lg Shoulder, 5/16"-18 Thread	91259A619	$1.27	12	$15.24
Motor mount and U-Bolt guard mounting hardware	McMaster	Double-Spring Tab Fastener, 5/16"-18 Thrd for Aluminum T-Slotted Framing Extrusion	47065T229	$1.46	8	$11.68
Main Sprocket Drive Shaft	McMaster	Fully Keyed Precision Drive Shaft with Certificate, 1" OD, 1/4" Keyway Width, 9" Length	8488T83	$27.10	2	$54.20
Battery terminal cover (black)	McMaster	Battery Terminal Cover, Lug, 2 & 1 AWG, Black (Negative)	69875K94	$2.00	5	$10.00
Battery terminal cover (red)	McMaster	Battery Terminal Cover, Lug, 2 & 1 AWG, Red (Positive)	69875K94	$2.00	5	$10.00
Roller chain guide	McMaster	Roller Chain Guide, Center Channel for ANSI #40/2040, 0.59" High, 8' Long	93095K18	$188.64	1	$188.64
U-Bolt mount	McMaster	Base Mount Shaft Support for 1/2" Shaft OD	6068K23	$25.99	2	$51.98
Plane guide UHMW tape	McMaster	High-Bond Wear-Resistant Slippery UHMW Tape, 1/2" Width x 15' Length, .022" Thick	7344A24	$8.03	2	$16.06
ANSI 40 Roller Chain	McMaster	Roller Chain, ANSI Number 40, 1/2" Pitch, 10' Long	6261K173	$45.40	1	$45.40
Fastening tabs for 15 Series Extruded Aluminum	McMaster	Double-Spring Tab Fastener, 5/16"-18 Thread for Aluminum T-Slotted Framing Extrusion	47065T229	$1.46	60	$87.60
End Caps for 10 Series Extruded Aluminum	McMaster	End Cap for 1" High Single Aluminum T-Slotted Framing Extrusion	47065T91	$1.20	10	$12.00
End Caps for 15 Series Extruded Aluminum	McMaster	End Cap for 1-1/2" High Single Aluminum T-Slotted Framing Extrusion	47065T87	$1.50	4	$6.00
1" Pillow Mount Bearings	McMaster	Lubricated Mounted Steel Ball Bearing, Set-Screw Lock, for 1" Shaft Diameter	5057N1	$82.14	4	$328.56
Wheel Adaptor Plate	Robot Marketplace	Machined Aluminum Wheel Hub	NPC-PH448	$20.00	2	$40.00
14" Flat Proof Wheel	Robot Marketplace	NPC-PT5306 14 inch flat-proof wheel	NPC-PT5306	$87.94	2	$175.88
Roller Chain Guide Tape	McMaster	3M VHB Foam Tape for Hard-to-Bond Surfaces, #4952, Adhesive Both Sides, 1" Wide x 5 Yard	76675A23	$36.53	1	$36.53
Secondary Sprocket	McMaster	Finished-Bore Sprocket with Hardened Teeth for #40 Chain, 1/2" Pitch, 30 Teeth, 1" Bore	2500T62	$57.24	1	$57.24
Scotch Extreme 1" x 3" Black Strip	Home Depot	Scotch Extreme 1" x 3" Black Strip	051131642546	$3.57	1	$3.57
Loctite 242 Blue Threadlocker	Home Depot	Loctite 242 Blue Threadlocker	079340242005	$6.47	1	$6.47
0.22in thick, 18x24 in Acrylic Sheet	Home Depot	0.22in thick, 18x24 in Acrylic Sheet	769125020316	$19.97	1	$19.97
0.093in thick, 18x24 in Acrylic Sheet	Home Depot	0.093in thick, 18x24 in Acrylic Sheet	769125010515	$9.78	4	$39.12
Adjustable Flag Bracket	Home Depot	Adjustable Flag Bracket	792723402253	$6.97	1	$6.97
					Total Mechanical Cost	**$3,292.93**

Table 2: Itemized Budget for AMPPS Electrical Components

Item	Vendor	Description	Part Number	Unit Cost	Quantity	Total Cost
AmpFlow Wheel Motor	AmpFlow	AmpFlow W43-500 Geared Wheel Motor w/10" Wheel	W43-500-SR-10B	$498.00	2	$996.00
12V, 22Ah Battery	Amazon	XS Power XP750 XP Series 12V 750 Amp AGM Supplemental Battery with M6 Terminal Bolt	XP750	$99.99	4	$399.96
ANL Fuse Holder	Amazon	E2 by Scoshe EWFH Single ANL Fuse Holder	EWFH	$6.05	1	$6.05
Manual ON/OFF Switch	Amazon	Blue Sea Systems 9003e e-Series Battery Switch Single Circuit ON/OFF	68180	$35.30	1	$35.30
8AWG Connectors for Wheel Motors	McMaster	Build-Your-Own Push-in Connector, Kit for 6 AWG, 75 Amps, Packs of 1 (2 Red, 2 White, 2 Black, 2 Green)	8026K2	$3.38	8	$27.04
8AWG Ring Terminals	McMaster	Standard Ring Terminal, Vinyl Insulated, 8 AWG, 5/16" Screw/Stud Size, Packs of 25	7113K223	$9.09	1	$9.09
Keeper 8ft x 1in Lashing Strap (2 pack)	Home Depot	Keeper 85243 8' x 2" Lashing Strap, 2 pack	85243	$7.97	1	$7.97
RoboteQ HDC2450 Brushed DC Motor Controller, Dual Channel, 150A, 50V, Encoder in, USB, CAN	Roboteq	HDC2450 Brushed DC Motor Controller, Dual Channel, 150A per Channel, 50V, with USB Input	HDC2450	$645.00	1	$645.00
Hook-Up Wire - Assortment (Solid Core, 22 AWG)	Sparkfun	Hook-Up Wire - Assortment (Solid Core, 22 AWG)	PRT-11367 RoHS	$16.95	1	$16.95
XS Power 580 Short Battery Post Adapters M6	Sonic Electronix	Pair of 12 Volt Short Brass Battery Terminal Post Adapters M6	XS Power 580	$10.99	4	$43.96
Bussmann ANN Very Fast-Acting Current Limiters ANN300	Summit Racing	ANN-300, 300 Amp Fast Acting Fuse	BSS-ANN300	$27.97	1	$27.97
Roboteq HDC2460S Brushed DC Motor Controller, Single Channel, 300A, 60V, Encoder in , USB	Roboteq	Roboteq HDC2450S Brushed DC Motor Controller, Single Channel, 300A per Channel, 60V max, with USB input	HDC2460S	$660.00	1	$660.00
Harsh Environment High-Amp Distribution Bar - 1 Circuit, 250 Amps @ 300 VAC, 4 Stud Terminals	McMaster	Harsh Environment High-Amp Distribution Bar - 1 Circuit, 250 Amps @ 300 VAC, 4 Stud Terminals	9290T17	$44.14	4	$176.56
Clear Cover for 9290T17 Harsh Environment High-Amp Distribution Bar	McMaster	Clear Cover for 9290T17 Harsh Environment High-Amp Distribution Bar	9290T29	$28.58	4	$114.32
Standard Ring Terminal - Vinyl Insulated, 22-18 AWG, 3/8" Screw/Stud Size	McMaster	Standard Ring Terminal - Vinyl Insulated, 22-18 AWG, 3/8" Screw/Stud Size (pack of 50)	7113K614	$11.74	2	$23.48
Gigavac GXNC14CB Normally Closed 350+ Amp 12-800 Vdc Contactor with 24 Vdc Coil	Gigavac	Gigavac GXNC14CB Normally Closed 350+ Amp 12-800 Vdc Contactor with 24Vdc Coil and 24 inch Coil Leads	GXNC14CB	$156.00	1	$156.00
Standard Heat-Shrink Ring Terminal8 AWG Wire Size, 3/8" Screw/Stud Size	McMaster	Standard Heat-Shrink Ring Terminal8 AWG Wire Size, 3/8" Screw/Stud Size	7036K74	$11.36	5	$56.80
Standard Heat-Shrink Ring Terminal22-18 AWG Wire Size, 3/8" Screw/Stud Size	McMaster	Standard Heat-Shrink Ring Terminal22-18 AWG Wire Size, 3/8" Screw/Stud Size	7036K63	$7.06	3	$21.18
Standard Ring TerminalVinyl Insulated, 8 AWG, 1/2" Screw/Stud Size	McMaster	Standard Ring TerminalVinyl Insulated, 8 AWG, 1/2" Screw/Stud Size	7113K716	$6.50	1	$6.50
Ultra-Flexible Wire 8 Gauge, Black, 10 ft long	McMaster	Ultra-Flexible Wire 8 Gauge, Black, 10 ft long	7479K13	$35.80	2	$71.60
45 Feet, 18 AWG stranded wire (three 15 ft rolls	Radio Shack	45 Feet, 18 AWG stranded wire (three 15 ft rolls)	2781226	$7.99	5	$39.95
SPST Rocker Switch	Radio Shack	SPST Rocker Switch	2750690	$3.49	3	$10.47
Noco 4 Channel Genius Charger	Amazon	NOCO Genius GEN4 40 Amp 4-Bank Waterproof Smart On-Board Battery Charger	B003JSLWWA	$320.95	1	$320.95

continued . . .

...Table 2 continued

Item	Vendor	Description	Part Number	Unit Cost	Quantity	Total Cost
T-Slot Cover, 6' Long for 1-1/2" High Aluminum T-Slotted Framing Extrusion	McMaster	T-Slot Cover, 6' Long for 1-1/2" High Aluminum T-Slotted Framing Extrusion	47065T4	$4.27	3	$12.81
DROK 10A/50W 9-32V 12V/24V to 5V Car DC Voltage Converter Regulator Power Supply,Waterproof	Amazon	DROK 10A/50W 9-32V 12V/24V to 5V Car DC Voltage Converter Regulator Power Supply,Waterproof	-	$15.49	1	$15.49
					Total Electrical Cost	**$4,426.40**

Table 3: Itemized Budget for AMPPS Tooling

Item	Vendor	Description	Part Number	Unit Cost	Quantity	Total Cost
Roller Chain Breaker	Amazon	T-Handle Roller Chain Breaker for Single & Double Strand Chain, ANSI NOS. 25-60	6051K15	$30.48	1	$30.48
Roller Chain Holder	Amazon	Jaw-Style Roller Chain Holder, T-Handle, for ANSI NOS. 40-80	6052K18	$55.97	1	$55.97
Roller Chain Tension Holder	McMaster	Jaw-Style Roller Chain Holder Knob, for ANSI NOS. 35-60	6052K14	$31.29	1	$31.29
Roller Chain Wear Guage	McMaster	ANSI Roller Chain Wear-Indicating Insert for ANSI NOS. 35-100	2370K2	$217.05	1	$217.05
3-D Print Material	Stratasys	PC Filament Canister Fortus 360/400mc	310-20100, "P	$395.00	1	$395.00
End Cutting Mill for counter-bore 10 series framing	McMaster	TIN-Coated High-Speed Steel 4-Flute Center Cut End Mill, 9/16" Mill Diameter, 1/2" Shank Diameter, 1-3/8" Length of Cut	8918A45	$35.04	1	$35.04
End Cutting Mill for counter-bore 15 series framing	McMaster	TIN-Coated High-Speed Steel 4-Flute Center Cut End Mill, 13/16" Mill Diameter, 3/4" Shank Diameter, 1-7/8" Length of Cut	8918A68	$45.60	1	$45.60
3-D Print Sheets	Stratasys	PC Filament Canister Fortus 360/400mc	310-00100	$395.00	1	$395.00
Stubby L-Wrenches	McMaster	Black-Oxide Inch Stubby Short-Arm L-Key Sets	6112A12	$18.54	1	$18.54
					Total Tool Cost	**$1,223.97**

Table 4: Itemized Budget for AMPPS Computing and Sensor Components

Item	Vendor	Description	Part Number	Unit Cost	Quantity	Total Cost
3652_0 - LCD Screen 2x20 - LCM2002J	Phidgets	LCD Screen 2x20	3652_0	$25.00	1	$25.00
1204_0 - PhidgetTextLCD Adapter	Phidgets	Phidget Adapter for LCD Display Screens	1204_0	$65.00	1	$65.00
1019_1 - PhidgetInterfaceKit 8/8/8 with 6 Port USB Hub	Phidgets	1019_1 - PhidgetInterfaceKit 8/8/8 with 6 Port USB Hub	1019_1	$125.00	1	$125.00
3919_0 - T5577 RFID Tag - PVC Disc 15mm	Phidgets	RFID Tag - 15mm PVC Disc	3919_0	$1.35	30	$40.50
1024_0 - PhidgetRFID Read-Write	Phidgets	RFID Read-Write Module	1024_0	$60.00	1	$60.00
1128_0 - MaxBotix EZ-1 Sonar Sensor	Phidgets	Phidgets Sonar Sensor	1128_0	$35.00	2	$70.00
1042_0 - PhidgetSpatial 3/3/3 Basic	Phidgets	Phidget Spatial 3/3/3 Basic Board	1042_0	$70.00	1	$70.00
3053_0 - Dual SSR Relay Board	Phidgets	Phidget Dual Solid State Relay Board	3053_0	$30.00	3	$90.00
3819_0 - Acrylic Enclosure for the 1204	Phidgets	Phidget Acrylic Enclosure for 1204 LCD Adapter	3819_0	$8.00	1	$8.00
3825_0 - Acrylic Enclosure for the 1024	Phidgets	Phidget Acrylic Enclosure for the 1024 RFID Reader	3825_0	$8.50	1	$8.50
3822_1 - Acrylic Enclosure for the 3053	Phidgets	Phidget Acrylic Enclosure for the 3053 SSR Relay Board	3822_1	$8.00	3	$24.00
3851_0 - Plastic Shell Enclosure for Spatials	Phidgets	Phidget Plastic Enclosure for 1042 Phidget Spatial 3/3/3	3851_0	$5.00	1	$5.00
Odroid-XU3	Ameridroid	Odroid XU-3 Single Board Computer	Odroid-XU3	$179.95	1	$179.95
DC Plug and Cable Assembly 5.5mm	Ameridroid	Odroid DC Plug and Cable Assembly 5.5mm	DC Plug and Cable	$1.45	1	$1.45
AC/DC 24V Red Green Yellow LED Lamp Industrial Tower Signal Light	Amazon	AC/DC 24V Red Green Yellow LED Lamp Industrial Tower Signal Light by Amico	a11080800ux 0057	$39.84	1	$39.84
RTC Battery for Oroid XU-3 Constant Power Supply	Ameridroid	RTC Battery	RTC Battery	$11.17	1	$11.17
3824_0 - Acrylic Enclosure for the 1019	Phidgets	3824_0 - Acrylic Enclosure for the 1019	3824_0	$10.00	1	$10.00
SanDisk Extreme Plus 32GB UHS-I/ U3 Micro SDHC Memory Card Up To 80MB/s With Adapter	Amazon	SanDisk Extreme Plus 32GB UHS-I/ U3 Micro SDHC Memory Card Up To 80MB/s With Adapter	SDSDQX-032G-AFFP-A	$34.48	1	$34.48
Monoprice 15-Feet USB 2.0 A Male to Mini-B 5pin Male 28/24AWG Cable with Ferrite Core (Gold Plated), White	Amazon	Monoprice 15-Feet USB 2.0 A Male to Mini-B 5pin Male 28/24AWG Cable with Ferrite Core (Gold Plated), White	108636	$5.94	1	$5.94
Logitech Gamepad F710 by Logitech	Amazon	Logitech Gamepad F710 by Logitech	F710	$38.48	1	$38.48
					Total Sensors/CPU Cost	**$912.31**

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

[1] Department of Defense. "Department of Defense dictionary of military and associated terms." 2014. [Online]. Available: http://www.dtic.mil/doctrine/new_pubs/jp1_02.pdf

[2] J. M. Abatti. "Small power: The role of micro and small UAVs in the future." Air War College, Maxwell AFB, Tech. Rep., 2005. [Online]. Available: www.dtic.mil/get-tr-doc/pdf?AD=ADA463472

[3] P. M. Miller, P. Manager, and M. K. Goodrich. "Mini, micro, and swarming unmanned aerial vehicles: A baseline study." Library of Congress, Washington, DC, Tech. Rep., 2006. [Online]. Available: www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA521374

[4] H. Geer and C. Bolkcom. "Unmanned aerial vehicles: Background and issues for Congress." Congressional Research Service, Washington, DC, Tech. Rep., 2005. [Online]. Available: www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA477738

[5] United States Naval Academy. "New warriors and new weapons: The ethical ramifications of emerging military technologies." in *McCain Conference*. Annapolis: United States Naval Academy Ethics Department, 2010. [Online]. Available: http://www.usna.edu/Ethics/_files/documents/publications/2010ExecutiveSummary-NewWarriors,NewWeapons.pdf

[6] C. Whitlock. "When drones fall from the sky: More than 400 have since 2001, U.S. military records show." Washington, DC, June 2014. [Online]. Available: http://www.washingtonpost.com/sf/investigative/2014/06/20/when-drones-fall-from-the-sky/

[7] C. C. Haddal and J. Gertler. "Homeland security: Unmanned aerial vehicles and border surveillance." Congressional Research Service, Washington, DC, Tech. Rep., 2010. [Online]. Available: http://fas.org/sgp/crs/homesec/RS21698.pdf

[8] Inspector General. "A review of remote surveillance technology along U.S. land borders." Department of Homeland Security, Tech. Rep. December, 2005. [Online]. Available: http://www.oig.dhs.gov/assets/Mgmt/OIG_06-15_Dec05.pdf

[9] "Unmanned systems integrated roadmap: FY2013-2038." Department of Defense, Tech. Rep., 2013. [Online]. Available: http://www.defense.gov/pubs/DOD-USRM-2013.pdf

[10] J. Gertler. "U.S. unmanned aerial systems." Congressional Research Service, Washington, DC, Tech. Rep., 2012. [Online]. Available: http://fas.org/sgp/crs/natsec/R42136.pdf

[11] S. Edwards. "Swarming and the future of warfare." Ph.D. dissertation, Pardee RAND Graduate School, 2005. [Online]. Available: http://www.rand.org/content/dam/rand/pubs/rgs_dissertations/2005/RAND_RGSD189.pdf

[12] E. Bonabeau and G. Theraulaz. "Swarm smarts." *Scientific American*, 2000. [Online]. Available: http://cognition.ups-tlse.fr/_guyt/documents/articles/33.pdf

[13] J. Arquilla and D. Ronfeldt. "Swarming and the future of conflict." 2000. [Online]. Available: http://www.rand.org/content/dam/rand/pubs/documented_briefings/2005/RAND_DB311.pdf

[14] W. D. Shannon. "Swarm tactics and the doctrinal void lessons from the Chechen Wars." M.A. thesis, Dept. National Security Affairs, Naval Postgraduate Sch., Monterey, CA, 2008.

[15] B. T. Clough. "UAV swarming? So what are those swarms, what are the implications, and how do we handle them?" Air Force Research Laboratory, Wright-Patterson AFB, Tech. Rep., 2002. [Online]. Available: http://www.researchgate.net/publication/235054087_UAV_Swarming_So_What_are_Those_Swarms_What_are_the_Implications_and_How_Do_We_Handle_Them

[16] A. Filippone. *Flight performance of fixed and rotary wing aircraft*. AIAA, 2006. [Online]. Available: http://books.google.com/books?hl=en&lr=&id=lTkiuG4RaFIC&pgis=1

[17] S.-W. Cheng. "Rapid deployment UAV." in *2008 IEEE Aerospace Conference*, no. 1488. Tainan: IEEE, Mar. 2008, pp. 1–8. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4526564

[18] R. B. Davis. "Applying cooperative localization to swarm UAVs using an extended Kalman filter." M.S. thesis, Dept. Computer Sci., Naval Postgraduate Sch., Monterey, CA, 2014.

[19] IHS Jane's. "AeroVironment Wasp AE UAV." *Jane's Unmanned Aerial Vehicles and Targets*, 2014. [Online]. Available: https://janes.ihs.com/

[20] IHS Jane's. "AeroVironment Puma AE UAV." *Jane's Unmanned Aerial Vehicles and Targets*, 2014. [Online]. Available: https://janes.ihs.com/

[21] IHS Jane's. "AeroVironment Raven UAV." *Jane's Unmanned Aerial Vehicles and Targets*, 2014. [Online]. Available: https://janes.ihs.com/

[22] IHS Jane's. "GA-ASI MQ-1B and RQ-1A Predator UAV." *Jane's Unmanned Aerial Vehicles and Targets*, 2013. [Online]. Available: https://janes.ihs.com/

[23] IHS Jane's. "Northrop Grumman/IAI MQ-5 and RQ-5 Hunter UAV." *Jane's Unmanned Aerial Vehicles and Targets*, 2014. [Online]. Available: https://janes.ihs.com/

[24] IHS Jane's. "Northrop Grumman RQ-4 Global Hawk UAV." *Jane's Unmanned Aerial Vehicles and Targets*, 2014. [Online]. Available: https://janes.ihs.com/

[25] IHS Jane's. "GA-ASI MQ-9 Reaper, Predator B and Mariner UAV." *Jane's Unmanned Aerial Vehicles and Targets*, pp. 1–13, 2013. [Online]. Available: https://janes.ihs.com/

[26] IHS Jane's. "AeroVironment Switchblade UAV." *Jane's Unmanned Aerial Vehicles and Targets*, 2014. [Online]. Available: https://janes.ihs.org

[27] IHS Jane's. "US Navy successfully launches XFC UAV from submarine." *Jane's Defense Weekly*, 2013. [Online]. Available: https://janes.ihs.com/

[28] IHS Jane's. "Boeing-Insitu ScanEagle UAV." *Jane's Unmanned Aerial Vehicles and Targets*, 2014. [Online]. Available: https://janes.ihs.com/

[29] IHS Jane's. "Textron Unmanned Systems RQ-7 Shadow UAV." *Jane's Unmanned Aerial Vehicles and Targets*, 2014. [Online]. Available: https://janes.ihs.com/

[30] IHS Jane's. "Textron Systems Aerosonde Mk 4.7 UAV." *Jane's Unmanned Aerial Vehicles and Targets*, 2014. [Online]. Available: https://janes.ihs.com/

[31] "Ritewing RC LLC - Radio controlled airplanes, RC aircraft, unmanned aerial vehicles." [Online]. Available: http://www.ritewingrc.com/home.html

[32] R. L. Davis. "Mechanical design and optimization of swarm capable UAV launch systems." M.S. thesis, Dept. Systems Eng., Naval Postgraduate Sch., Monterey, CA, 2015.

[33] Department of Defense. "Defense acquisition guidebook." 2013. [Online]. Available: https://dag.dau.mil/

[34] Department of Defense. "Joint capabilities integration and development system." 2012. [Online]. Available: https://dap.dau.mil/policy/Documents/2012/3170_01.pdf

[35] Department of Defense. "Manual for the operation of the joint capabilities integration and development system." 2012. [Online]. Available: https://dap.dau.mil/policy/Documents/2012/JCIDSManual19Jan2012.pdf

[36] C. Ragsdale, *Spreadsheet modeling and decision analysis*, 5th ed. Mason, OH: Thompson South-Western, 2008.

[37] M. R. Middleton, *Decision analysis using microsoft excel*. San Francisco: University of San Francisco, 2006.

[38] G. Viera, J. Webb, D. Cummings, P. Livesay, R. Davis, J. Ng, and A. Abulfateh, "Rapid unmanned aerial vehicle launch engine design capstone report," M.S. Capstone Report, Dept. Systems Eng., Naval Postgraduate Sch., Monterey, CA, 2014.

[39] "ROS.org | Powering the world's robots." [Online]. Available: http://www.ros.org/

[40] "Logitech F710 wireless gamepad." [Online]. Available: http://gaming.logitech.com/en-us/product/f710-wireless-gamepad

[41] "Oregon Scientific - professional weather center." [Online]. Available: http://www.oregonscientific.com/us/en/Professional-Weather-Center-WMR200A-P

[42] "WeeWx: Linux weather software." [Online]. Available: http://weewx.com/

[43] "RFID primer - Phidgets support." 2013. [Online]. Available: http://www.phidgets.com/docs/RFID_Primer

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California